

# 68000

**\$2.95<sub>USA</sub>**

Australia  
Singapore  
Malaysia

A \$ 4.75  
S \$ 9.45  
M \$ 9.45

New Zealand  
Hong Kong  
Sweden

NZ \$ 6.50  
H \$23.50  
30>SEK

## MICRO JOURNAL

**VOLUME VI ISSUE IV • Devoted to the 68XX User • April 1984**

"Small Computers Doing Big Things"

SERVING THE 68XX USER WORLDWIDE



PHOTO CREDIT: NASA

# WE DON'T PLAY GAMES



## **X-12+** A SERIOUS COMPUTER IN A DESKTOP PACKAGE

**Multiprocessor Technology - Combination of 8, 16 and 32 bit types**

**1.0 Megabyte Memory - Insures no limitation on programs**

**"Winchester" Disk System - Fast response, large storage capacity**

**UniFlex<sup>®</sup> Operating System - The standard of comparison**

**Hardware Floating Point - Unmatched speed in a small system**

**Up to Three Terminals - Instant expansion**

\*Trademark of Technical Systems Consultants



SOUTHWEST TECHNICAL PRODUCTS CORPORATION  
219 W. RHAPSODY  
SAN ANTONIO, TEXAS 78216

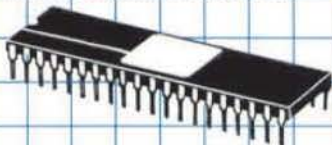
(512) 344-0241



# MICROWARE'S OS-9 IS NUMBER ONE.

## OS-9 NOW HAS THE LARGEST USER COMMUNITY

More users now run OS-9 on their 6809 computers than all other operating systems combined. This outstanding success story was no accident — it's due to OS-9's technical excellence backed up by outstanding Microware support. OS-9's Unix-type architecture and totally modular design gives your computer more power and versatility. OS-9 also gives you more possibilities for customization so you can tailor your system exactly to your needs. And aren't flexibility and performance the reasons you chose a 6809 computer to begin with?



## OS-9 HAS BEEN CHOSEN BY OVER 50 6809 SYSTEM MANUFACTURERS

OS-9 is now offered as a standard operating system by almost every 6809 system manufacturer, and has been designed into an amazing variety of dedicated systems and products including personal and business computers, process control systems, data and telecommunications systems, and more. In all, over 50 companies and organizations have

obtained OS-9 distribution licenses including such well-known names such as General Motors, NASA, Fujitsu, Western Electric, Motorola, Sykes Datatronics, Eastman Kodak, Thomson-CSF, and Tandy Corp.

## OS-9 GIVES YOU A SOFTWARE BASE TO BUILD ON

Whatever your application, OS-9 speaks your language! Microware offers BASIC09, an Extended/Structured Basic, a complete C Compiler, a full ISO Pascal Compiler, the ANSI Standard CIS Cobol Compiler, plus Relocatable Macro Assembler. These high performance programming languages are all fully implemented and deliver unmatched performance and outstanding features. Additionally, OS-9 compatible applications packages such as word processors, screen editors, spreadsheets, business software, and utilities are offered by a rapidly growing number of third-party software houses.

## PLUS OUTSTANDING MICROWARE SUPPORT: WE KEEP IN TOUCH WITH YOU

Even when you have the best software and documentation, there can be times when you need questions answered. That's why Microware is committed to giving OS-9 users the best possible personalized service. Here are some

of the ways we deliver solid support:

- A Software Support Hotline for direct access to our technical staff
- "Pipelines", our free quarterly newsletter
- OS-9 User Seminars, the annual OS-9 community gathering
- a liberal update policy for new releases

Microware does business on a person-to-person basis. When you call you'll find yourself speaking with someone who's both knowledgeable and genuinely interested in helping.

## YOU CAN COUNT ON OS-9 NOW AND IN THE FUTURE

Microware is not standing still — we're firmly committed to continuing support for the 6809 and we will continue to introduce exciting new software products for the 6809. We will soon announce OS-9/68000 and programming languages for the 68000 which will be upward compatible with 6809 versions, so if and when you are ready for the 68000 your OS-9 software can go with you.



**MICROWARE.**

**Microware Systems Corporation**  
Box 4865 • Des Moines, IA 50304  
515-279-8844 Telex 910-520-2535

OS-9 and Basic09 are trademarks of Microware and Motorola.  
Unix is a trademark of Bell Labs.

## OS-9: BETTER BY DESIGN

# '68'

Portions of the text for 68 MICRO JOURNAL was prepared using the following furnished hard/software.

## COMPUTERS-HARDWARE

Southwest Technical Products  
219 W. Rhapsody  
San Antonio, Texas 78216  
S09-5/8 DMF disk-CDS1-8212W-Sprint 3 Printer

GIMIX Inc.  
1337 West 37th Place  
Chicago, IL 60609  
Super Mainframe-OS9-FLEX-Assorted Hardware

## EDITORS-WORD PROCESSORS

Technical Systems Consultants, Inc.  
111 Providence Road  
Chapel Hill, NC 27514  
FLEX-Editor-Processor

Great Plains Computer Company, Inc.  
PO Box 916  
Idaho Falls, ID 83401  
STYLO-Mail Merge

### Editorial Staff

Don Williams Sr.	Publisher
Larry E. Williams	Executive Editor
Tom E. Williams	Production Editor
Robert (Bob) Nay	Color Editor

### Administrative Staff

Mary Robertson	Office Manager
Joyce Williams	Accounting
Carolyn Williams	Subscriptions
Penny Williams	File Management

### Contributing Editors

Ron Anderson  
Norm Commo  
Peter Dibble  
Dr. Theo Elbert  
William E. Fisher  
Dr. E.M. Pass

### Special Technical Projects

Clay Abrams K6AEP  
Tom Hunt

## CONTENTS

Vol. VI, Issue IV

APRIL '84

FLEX USER NOTES.....	8	Anderson
OS9 USER NOTES.....	11	Dibble
C USER NOTES.....	16	Commo
Where Is The 6800 .....	27	Staff
Programming In C: A Review.....	29	Foster
The GIMIX III System.....	29	Staff
Remote Pulse Width.....	31	Craig & Zolowski
James Bellomo Dies.....	34	
Day Command for FLEX-9.....	34	Mills
Bit Bucket.....	39	
OS-9 User Group News Continued.	39	Part II
68000 SAGE System.....	42	Lord
Budget.....	45	Craig
Classifieds.....	49	

# MICRO JOURNAL

## Send All Correspondence To:

Computer Publishing Center  
68 MICRO JOURNAL  
5900 Cassandra Smith  
PO Box 849  
Hixson, TN 37343  
615 842-4600

Copyrighted 1984 by  
Computer Publishing Inc. (CPI)

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

### SUBSCRIPTION RATES

USA

1-Year \$24.50 2-Years \$42.50 3-Years \$64.50

FOREIGN

See Page 60

### Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors full name, address, date and telephone number. It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands ONLY should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the ,pg page command, we print edited text files in continuous text.

All articles submitted on diskettes should be in TSC FLEX" format, either FLEX2 6800, or FLEX9 6809 any version.

If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). Please use a dark ribbon!

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less \$7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.



# GIMIX HAS THE 6809 SYSTEM TO SUIT YOUR NEEDS

## HARDWARE

All systems feature the **GIMIX CLASSY CHASSIS**; with a ferro-resonant constant voltage power supply, gold plated bus connectors, and plenty of capacity for future expansion.

Static **RAM** and double-density **DMA** floppy disk controllers are used exclusively in all systems.

All systems are guaranteed for 2 MHz operation and include complete hardware and software documentation, necessary cables, filler plates, etc.

Systems are assembled using burned-in and tested boards, and all disk drives are tested and aligned by **GIMIX**.

You can add additional components to any system when ordering, or expand it in the future by adding **RAM**, **I/O**, etc.

**GIMIX** lets you choose from a wide variety of options to customize your system to your needs.

## OS-9 GMX III/FLEX SYSTEMS (#79)

The #79 super system now includes (in addition to the above): the **GMX 6809 CPU III**, a **256K CMOS Static RAM Board (#72)**, and a **3-port Intelligent Serial I/O Processor (#11)**.

The **GMX 6809 CPU III** can perform high-speed **DMA** transfers from memory to memory and uses memory attributes and illegal instruction trapping to protect the system and users from program crashes. If a user program crashes, only that user is affected; other users are unaware of the problem.

The **3-Port Intelligent Serial I/O Board (#11)** significantly reduces system overhead by handling routine **I/O** functions; freeing the host CPU for running user programs. This improves overall system performance and allows user terminals to be run at up to 19.2K baud.

with dual 40 track DSDD drives	\$5998.79
with dual 80 track DSDD drives	\$6198.79
with #88 dual 8" DSDD drive system	\$7698.79
with #90 19MB Winchester subsystem and one 80 track	\$8898.79
with a 47MB Winchester subsystem and one 80 track	\$10,898.79
with a 47MB plus a 6MB removable pack Winchester subsystem and one 80 track drive	\$12,398.79

**TO ORDER BY MAIL:** SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add \$5 handling if order is under \$200.00. Foreign orders add \$10 handling if order is under \$200.00. Foreign orders over \$200.00 will be shipped via **Emery Air Freight COLLECT**, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the **Continental Illinois National Bank of Chicago**, 231 S. LaSalle Street, Chicago, IL 60603, account #73-32033.

**BASIC-09** and **OS-9** are trademarks of Microware Systems Corp. and **MOTOROLA**, Inc. **FLEX** and **UniFLEX** are trademarks of Technical Systems Consultants, Inc. **GIMIX**, **GHOST**, **GMX**, **CLASSY CHASSIS**, are trademarks of **GIMIX**, Inc.

## SOFTWARE

All **OS-9/FLEX** systems allow you to software select either operating system.

Also included is the **GMXBUG** monitor and, in systems with 128K or more of **RAM**, **GMX-VDISK** for **FLEX**.

All **GIMIX OS-9** systems include **Microware's Editor, Assembler, Debugger, Basic09**, and **Runb**; and the **GMX** versions of **RMS** and **DO** for **OS-9**.

All **GIMIX** versions of **OS-9** can read and write **RS color computer format OS-9** disks, as well as the **Microware/GIMIX standard format**.

New and exclusive with **OS-9 GMX III** systems is the **GMX OS-9 Support ROM**, a monitor for **OS-9** that includes memory diagnostics and allows the system to boot directly from either hard disk or floppy.

A wide variety of languages and other software is available for use with either **OS-9** or **FLEX**.

## OS-9 GMX I / FLEX SYSTEMS #49

The #49 systems include 64KB static **RAM**, #05 **CPU**, #43 2 port serial board.

with dual 40 track DSDD drives	\$3998.49
with dual 80 track DSDD drives	\$4198.49
with #88 dual 8" DSDD drive system	\$5698.49
with #90 19MB Winchester subsystem and one 80 track	\$6898.49

## OS-9 GMX II / FLEX SYSTEMS #39

The #39 systems include 128KB static **RAM**, #05 **CPU**, #43 2 port serial board.

with dual 40 track DSDD drives	\$4498.39
with dual 80 track DSDD drives	\$4698.39
with #88 dual 8" DSDD drive system	\$6198.39
with #90 19MB Winchester subsystem and one 80 track	\$7398.39

**GIMIX DOES NOT GUARANTEE PERFORMANCE OF ANY GIMIX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS PRODUCT.**

**EXPORT MODELS: ADD \$30 FOR 50Hz. POWER SUPPLIES.**

**GIMIX, Inc.** reserves the right to change pricing, terms, and products specifications at any time without further notice.

ALL PRICES ARE **F.O.B. CHICAGO**

Contact **GIMIX** for price and availability of **UniFLEX** and **UniFLEX GMXIII** Systems.

**NOTE** on all drive systems: Dual 40 track drives have about 700KB of formatted capacity; dual 80's about 1,400KB; dual 8" about 2,000KB. The formatted capacity of hard disks is about 80% of the total capacity.

## Want to expand your system to a megabyte of Static RAM and 15 users?

Simply add additional memory and **I/O** boards. Your **GIMIX** system can grow with your needs. Contact us for a complete list of available boards and options.

#72 256KB CMOS STATIC RAM board	
with battery back up	\$1898.72
#64 64KB CMOS STATIC RAM board	
with battery back up	\$528.64
#67 64KB STATIC RAM board	\$478.67
#11 3 port Intelligent serial I/O board	\$498.11
#43 2 port serial I/O board	\$128.43
#42 2 port parallel I/O board	\$88.42
#95 cable sets (1 needed per port), specify board	\$24.95

## TRADE UP YOUR CoCo!

**GIMIX** will allow you up to \$1100.00 credit toward the purchase of any **GIMIX** system when you trade-in your working **Color Computer**, peripherals, and original software. The trade-in value is limited to 110% of the **RADIO SHACK™** list price at the time your order is placed. You pay the freight. This offer is good only in the Continental U.S.; is limited to the first 100 orders; and expires on 9/30/84. Only one trade-in per customer.

**GIMIX** inc.

1337 WEST 37th PLACE  
CHICAGO, ILLINOIS 60609

(312) 927-5510 • TWX 910-221-4055



©1984 GIMIX, INC. 4-84

# Microware presents 4 new OS-9 software packages.

## 1 LEVEL II PRINT SPOOLING SYSTEM

This versatile package gives your OS-9 Level Two System a complete print spooling management capability for time-sharing applications. Features of the spooling system are:

- Handles up to seven independent spooling devices and queues with "print on first available device" feature.
- Prints large block header pages between listings with date, time, user name and job name.
- Multiple listing copy option.
- Complete forms change capability for each job and device.
- Prints formatted or unformatted listings.
- Status command displays print queues and status.
- User can kill or change priority of queued jobs.

Available only for OS-9 Level Two Systems.

**Suggested List Price: \$150.00 Manual Only: \$15.00**

## 2 RMA RELOCATABLE MACRO ASSEMBLER

At last — a full feature relocatable macro assembler and linkage editor for OS-9. RMA permits sections of assembly language programs to be independently assembled to "relocatable object files". The linkage editor takes any number of program sections and/or library sections and combines them into a single executable OS-9 memory module. Global data (including indexed and direct addressing modes) and program references are automatically resolved in the process. The macro facility permits commonly used statement sequences to be defined, then used within the program with appropriate parameter substitution. RMA also supports conditional assembly and library source files.

**Suggested List Price: \$200.00 Manual Only: \$20.00**

## 3 OS-9 FILE HANDLER TOOLBOX

Introducing a special toolbox for OS-9 users who do a lot of file manipulation! A collection of 12 useful OS-9 command

programs; Most can be used as "filters" using OS-9 pipeline facilities. Included are:

**D** — unformatted directory listing with "wild card" matching

**Compress** — does character compression on text files.

**Expand** — restores a "compressed" file to the original state.

**Split** — breaks a file into smaller files.

**Space** — indents lines with optional spacing between lines.

**Code** — decodes any key on a keyboard to hex.

**Osrt** — quick sort for small files, directories, etc.

**Pr** — versatile formatted file printing utility.

**Tr** — transliterates text pattern to substitution pattern.

**Grep** — searches file for a pattern and prints matching lines.

**Xmode** — same "tmode" except changes are made to the device descriptor.

**Count** — counts words, lines, or characters within a text file.

**Suggested List Price \$85.00**

## 4 ENTERTAINMENT PACK I

A collection of games and other interesting programs that are not only entertaining but serve as good instructional examples of Basic09 programming techniques. All programs include complete Basic09 source files and can be easily edited to run on standard alphanumeric or graphics terminals.

**Blkjak** — A Vegas-rules blackjack game.

**Clk** — graphical display of a wall clock on your terminal.

**Dogs** — Greyhound racing with simulated graphics.

**Eltza** — Basic09 version of the famous artificial intelligence simulation of natural language dialogue with a psychiatrist.

**Haiku** — Program that creates original "haiku" prose.

**Quest** — a mini-"Adventure" game.

**Rats** — find your way out of a computer-generated maze — from a rat's point of view.

**Towers** — a graphical display of the solution to the "Tower of Hanoi" puzzle.

**Suggested List Price: \$85.00**



**MICROWARE.**

**Microware Systems Corporation**

P.O. Box 4865 • Des Moines, IA 50304

515-279-8844 • Telex 910-520-2535

OS-9 and Basic09 are trademarks of Microware and Intel.  
Unix is a trademark of Bell Labs.



# FLEX™ USER NOTES THE 6800-6809 BOOK

By: Ronald W. Anderson  
As published in 68 MICRO JOURNAL™

The publishers of 68 MICRO JOURNAL are proud to announce the publication of Ron Anderson's **FLEX USER NOTES**, in book form. This popular monthly column has been a regular feature in 68 MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. Now all his columns are being published, in whole, as the most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

As a **SPECIAL BONUS** all the source listing in the book will be available on disk for the low price of: FLEX™ format only — 5" \$12.95 — 8" \$16.95 plus \$2.50 shipping and handling, if ordered with the book. If ordered separately the price of the disks will be: 5" \$17.95 — 8" \$19.95 plus \$2.50 shipping and handling.

Listed below are a few of the **TEXT** files included in the book and on diskette.

All TEXT files in the book are on the disks.

LOGO.C1  
MEMOVE.C1  
DUMP.C1  
SUBTEST.C1  
TERMEN.C2  
M.C2  
PRINT.C3  
MODEM.C2  
SCIPKG.C1  
U.C4  
PRINT.C4  
SET.C5  
SETBAS1.C5

File load program to offset memory — ASM PIC  
Memory move program — ASM PIC  
Printer dump program — uses LOGO — ASM PIC  
Simulation of 6800 code to 6809, show differences — ASM  
Modem input to disk (or other port input to disk) — ASM  
Output a file to modem (or another port) — ASM  
Parallel (enhanced) printer driver — ASM  
TTL output to CRT and modem (or other port) — ASM  
Scientific math routines — PASCAL  
Mini-monitor, disk resident, many useful functions — ASM  
Parallel printer driver, without PFLAG — ASM  
Set printer modes — ASM  
Set printer modes — A-BASIC  
(And many more)

\*\*Over 30 TEXT files included in ASM (assembler) — PASCAL — PIC (position independent code) TSC BASIC-C, etc.

NOTE: .C1, .C2, etc. = Chapter 1, Chapter 2, etc.

This will be a limited run and we cannot guarantee that supplies will last long. Order now for early delivery.

Foreign Orders Add \$4.50 S/H

Softcover — Large Format

Book only: **\$7.95** + \$2.50 S/H

With disk: 5" **\$20.90** + \$2.50 S/H

With disk: 8" **\$22.90** + \$2.50 S/H

See your local S50 dealer/bookstore or order direct from:

**Computer Publishing Inc.**  
**5900 Cassandra Smith Rd.**  
**Hixson, TN 37343**  
**(615) 842-4601**

™FLEX is a trademark of Technical Systems Consultants



# !!! FREE !!!

Published Monthly by Computer Publishing Inc., Hixson, TN.

**\$1.95**



Bulk Rate  
U.S. Postage  
PAID  
Chattanooga, TN  
Permit No. 357

## Color Micro Journal

The Color Computer Monthly Magazine

\$1.95 per issue Vol. 1, Issue 2 October, 1983

### THIS 'N THAT

The **BIG NEWS** this month is that OS-9 has finally arrived for the Color Computer. The **ASTOUNDING** part of the Radio Shack OS-9 Package, besides the price, is the **documentation**. You 'Old Time Radio Shack Followers' will not believe what you see. Jon Shirley has been telling us that the main reason for the "lack" of documentation with a lot of their products was the restrictions placed on releasing that information by **Microsoft**. I

One of the "Operating Systems of the Future" is **now available** for the "little old Color Computer"; OS-9. Freely translated, OS-9 means "Operating System for the 6809" (OS-9 is now being written for the 68000, also). Since it is fairly obvious that UNIX and "UNIX-Type" Operating Systems will be running on just about every computer to come out in the next few years, a whole new language is beginning to appear on the horizon.

### OS-9 on the COLOR COMPUTER

#### Color Computer OS-9, the Package

We had been running a preliminary release of OS-9 on the Color Computer for a few weeks, and received the "Official Radio Shack" version for Review a couple of days ago. To put it mildly, this package is **IMPRESSIVE**. For \$69.95 (Radio Shack Catalog Number 26-3838), you receive a 9 1/2" x 7 5/8" x 2" package containing 4

## FREE SAMPLE ISSUE

### 1-800-338 6800

MON.-FRI. 9-5 E.S.T.

USA-\$12.50 per year. Canada& Mexico-\$19.50 per year

Surface Foreign-\$24.50 per year. Airmail Foreign-\$48.50 per year

## Color Micro Journal™

™ Color Micro Journal is a trademark of Computer Publishing Inc.

5900 Cassandra Smith Rd.

Hixson, TN. 37343



# TMP (Total Management Planning) OS-9 SOFTWARE

THESE OFF-THE-SHELF APPLICATION PACKAGES HAVE CHANGED THE PICTURE  
FOR PROFESSIONAL SOFTWARE USERS

*Here are the comprehensively supported OS-9 application software packages that discerning professionals are selecting today. Each one runs stand-alone or in combination with other TMP packages.*

**TMP/CALC** is a new-generation "Electronic Spreadsheet" package that goes a quantum leap beyond ordinary Calc software. Superior speed, and extra features like Dynamic Calculations and Dynamic Overlay, mean you can change a cell *anywhere* on the spreadsheet and get automatic updates wherever they belong. No more "wrap around" since output to the printer is formatted. "Help" screens prompt without erasing data, and the rows and columns on any worksheet are limited only by available memory.  
ONLY \$250

**TMP/Manager** is a structured database manager, and is the ultimate driving force behind any total Management/Marketing Planning system. Its built-in select/sort/merge module — and optimized machine code — makes it super-fast, and provides unusually nimble data manipulation characteristics. Other outstanding features are detailed prompting, easy identification of data fields, and an excellent security system.  
ONLY \$500

**TMP FreeForm** fills the void between a structured DBMS and word processing. It's an "electronic index card" package with an endless list of valuable applications such as parts cataloging, look-up inventories and listings of any kind. Type anything on up to 32,000 electronic "cards" of up to 9 pages each that reside in "file drawers" (databases). Up to 13 key-words can call each page.  
ONLY \$150

**TMP/Front-End** integrates any or all of the packages described above, and integrates TMP with word processors, BASIC and other high-level languages. NO CHARGE with TMP/Manager



**TMP PACKAGES ARE EXCEPTIONALLY FRIENDLY. EACH COMES WITH MULTI-MEDIA TRAINING AND OUTSTANDING DOCUMENTATION**

**AUDIO CASSETTE TRAINING TAPES** are provided with each TMP software package. These crisp, professionally produced training aids take a new user

through "hands on" exercises that instantly build confidence, and minimize reference to manuals.

**CLEAR DOCUMENTATION** is specially prepared for each TMP package. Remarkably readable because they are conversational in style and logically organized, this series of manuals further assures quick user comfort and productivity with TMP software. Includes sections on how to best utilize the system: integrating user-written software with TMP packages, and other bonus topics.

**VIDEOTAPE PROGRAMS** are available to TMP Dealers. They dramatically take the viewer into real business environments for a close look at how a TMP software package is utilized. Video programs can be personalized for specific organizations. Programs may be specified for most popular tape formats.

**THE BOTTOM-LINE IS THIS:** Software is only as effective as its training and documentation. TMP packages are *incomparable* in both areas.

**RUN TMP IN A SMOKE SIGNAL CHIEFTAIN™ COMPUTER TO EXPERIENCE THE KIND OF SPEED AND POWER YOU CAN EASILY AFFORD TODAY**



Endurance-Certified Chieftains are consistently among the fastest computers in CPU and I/O speeds according to the widely respected BENCHMARK REPORTS. That, combined with almost legendary reliability, makes Chieftains the logical choice in computers using OS-9 operating systems.

Configurations range from floppy-disk systems to multi-user, multi-tasking Winchester hard disk systems with tape back-up; performance approaching mainframes at prices you can live with.

TMP software packages for most OS-9 systems and Chieftain computers are offered exclusively from Smoke Signal. Inquiries from non-Smoke Signal dealers are invited. Call (213) 889-9340.

**TMP** T.M.



**SMOKE SIGNAL**  
Chieftain Computers

31336 Via Colinas • Westlake Village, CA 91362  
(213) 889-9340

# Flex User Notes

Ronald W. Anderson  
3540 Sturbridge Court  
Ann Arbor, MI 48105

## Human Nature

This is not being written to make anyone look foolish. If it were, I would be a prime candidate for being the foolish one. I'd like to go into a little story here that I think reveals something about how we tend to stick with something with which we are comfortable, rather than risk trying "the easy way" that we don't know as much about.

You all have heard my tale of how I got started in computing with a KIM-1 single board computer back in 1977, so I won't go into the whole story. That system had a Hexadecimal keypad and six seven segment displays. To write a program for it, it was necessary to use the program card to translate the instructions into the proper Hex machine codes, calculate branch and offset values in Hex, and then enter the program via the keypad. I became fairly proficient at doing it that way. About a month after starting with the KIM-1, I ordered a SWTPc system. It arrived a few months later, and I learned the 6800 instruction set and started coding in machine code again, with good success.

Along with the system came Motorola's Co-Res assembler, a combination editor and assembler package. I was hesitant to use it because of all the assembler directives and pseudo operation codes I would have to learn. When I expressed my fears to the local computer store owner, he laughed and told me that I had been doing it the hard way for several months, and that a few hours with the assembler would convince me that my fears were ungrounded. Of course he was correct. Using the assembler turned out to be so much easier than hand coding, that I was able to go on to bigger and better programs with much less overall effort.

Recently, I have found myself consulting for a small company in the area of software. I'm pushing that company to start using a compiler. The reasons I get back for not switching, sound just like my objections to starting to use the assembler so long ago. The customer feels comfortable with assembler. It is easy to make changes, using assembler. "Besides, the little box we are programming has to communicate with another computer via a couple of parallel interfaces which will require drivers in assembler anyway" (customer's words, and not true as you will see below).

I still find myself guilty of resisting change for reason of familiarity. I have some hardware sitting here (have had it for two or three months) with which I could be up and running OS-9 with a little work on my hardware. I don't get around to it because I am so familiar with FLEX that I am comfortable, and don't want to expend the effort to learn a "new operating system". I look at the 68000 and make excuses to resist starting to get a system together to use it. I probably will get around to that when I find an application in which the 6809 simply can't do the job. I think it is clear that we all resist change for the reason of familiarity, even if the change might be in the direction of doing it the easy way.

## PL/9

One of my readers who lives in the Detroit area, called me the other day and said that I should give some programming examples that show why I think PL/9 is so ideal for my programming. Maybe the best way would be to show that you can write an I/O driver for something like a serial port (6850) without resorting

to any assembler code at all. Suppose we have a parallel port at address \$E030, just for an example. Let's say we want to connect a modem to that port and write a PUTMOD and GETMOD function that correspond with PUTCHR and GETCHR of FLEX. First, somewhere near the top of our PL/9 program we define the port and give its data and control registers names:

```
AT $E030 BYTE MOCONTROL,MODATA;
```

This has created two variables, one called MOCONTROL, a single byte at \$E030, and a second called MODATA at the next address \$E031. Of course, having a port defined, doesn't make it work, so we have to initialize it. Here is a procedure to do that:

```
PROCEDURE PORTINIT;  
  MOCONTROL = $3;  
  MODATA = $15;  
ENDPROC;
```

Now a call in the program to this procedure will initialize the port. All that remains is to write the PUTMOD and GETMOD functions:

```
PROCEDURE PUTMOD (BYTE CHAR);  
  REPEAT UNTIL MOCONTROL AND 2 <> 0;  
  MODATA = CHAR;  
ENDPROC;
```

```
PROCEDURE GETMOD;  
  REPEAT UNTIL MOCONTROL AND 1 <> 0;  
ENDPROC BYTE MODATA;
```

The definition of the procedure PUTMOD declares that a character will be passed to it by the main program. It first loops until the transmit buffer empty flag is set, and then assigns the value of the character to the variable MODATA, thus outputting the character. Note that the loop does nothing but repeat the test for exit from the loop. In other words, there are no statements included in the REPEAT UNTIL loop, just the test at the end. GETMOD waits for the receive buffer full flag, and then returns the value of MODATA to the calling procedure.

Tests for true and false values have an implied <> 0 for true. "UNTIL MOCONTROL AND 1" therefore is the same as "UNTIL MOCONTROL AND 1 <> 0". Note that the AND is a bitwise AND function. A logical AND is preceded by a dot ".AND". In the main program the two procedures would be used as follows:

```
PORTINIT;  
PUTMOD('A'); /* PUTMOD(65) WOULD DO AS WELL */  
CHAR = GETMOD;
```

Of course the PORTINIT call must precede the use of either PUTMOD or GETMOD. PUTMOD requires a parameter to be passed to it. The parameter should be a character. GETMOD returns a character, which must be assigned to a variable. I should add that in PL/9 BYTE and CHAR are synonymous. A character and a byte are both treated as 8 bit values. Integers are 16 bit values, and REALS are four bytes.

PL/9 allows you to get "closer to the machine" if you wish. It has named the registers of the 6809 as variables. ACCA = 'T'; does the same thing as LDAA #T would in an assembly language program. You can embed assembler code in the middle of a PL/9 program if you wish. PL/9 treats each line as a complete entity. After a line of code is executed, the compiler makes no assumptions about the contents of the accumulators. You can do some assembler manipulation and then return to PL/9 code on the next line. PL/9 does use the stack pointer and the Y register internally, and if you use them in your assembler code, they must be saved and restored before proceeding to the next line of PL/9 code. Parameters are passed on the system stack, to a procedure, and the calling



procedure or main program cleans up the stack after the return. Procedures that return values (functions) return the value in the B register (byte), D register (integer) or D and X registers (real).

As in Pascal, a function can only return a single value. Procedures that need to return multiple values (such as rectangular to polar coordinate conversion functions) can do so by means of the calling program passing pointers to the place where the results should be placed (just as in Pascal).

```
PROCEDURE RECT (REAL R, THETA, .X, .Y);
  X = R * COS(THETA);
  Y = R * SIN(THETA);
ENDPROC;
```

The main program passes the radius and angle (polar coordinate form of a vector) and pointers to the two real variables where X and Y (rectangular form) are to be placed. The procedure does the calculations, using the "pointed at" locations to store the results. The calling program would look like:

```
RECT (AMOUNT, ANGLE, .XVAL, .YVAL);
```

The above variables would have to have been defined as REAL variables in the program. The dot "." before the variable name indicates that the ADDRESS of the variable is to be passed to the procedure RECT, and not the contents of the variable. Note that all pointers are 16 bits, and as such the POINTER is an integer. In the definition of RECT above, the word REAL indicates that they are pointers to REAL variables. The type used is the type of the variable to which the pointer is pointing. One other point is to be made. Variables passed to a procedure are handled the same as local variables in that procedure, and as such their values or pointers to them may be passed to other procedures. Procedures in PL/9 may not be nested. That is, a procedure may not be defined within another procedure. However, any procedure may call any procedure declared before it, so that when the program is run, it may get several procedures (subroutines) deep at times. MAIN may call PSTRNG, which may call PUTMOD, etc.

Well, that is certainly enough for one session on PL/9.

#### READER RESPONSE

I continue to get comments from readers on the compiler - assembler debate. This is from Dr. Delbert Franz who has been a consultant in civil engineering and has recently gone into the software consulting area as well.

"Assembly language programming was on the wane until the micros came along with their severely limited memory and speed. Now memory is cheaper and speeds are higher so that the second death of assembly programming can begin. I have no interest in being tied to a specific processor by extensive assembler code. Often the program spends most of its time in only 10% of the code. After the fact optimization is often successful if the program runs too slowly. In my case assembler would make little difference because I primarily work with number crunching programs. Most of the time is spent in the floating point processor."

I wrote the above yesterday. Today I received the next installment of words from Dan Farnsworth, in response to my last response to his last.... You get the idea. I quote Dan's letter directly.

"I generally agree with your rebuttal and only take issue with your use of the "secret code" phrase. If you have the source code to the subroutines they are no longer a secret or a mystery, and if you don't have the source code you couldn't use them anyway. The code that is in my runtime package isn't much more

complicated than OUTCH, or INCH, or POATA. It is just very convenient to JSR CPDATC and output a CR & LF, then print the message at [X] and another CR & LF. I think you described my system very well when you said that I had written my own compiler, and I might add that the compiler is controlled by assembly language instead of a Syntax.

"One of the most common structures in HLLs is the Loop. This is the FOR - NEXT statement in BASIC. In assembler this could be written as:

```
LOOP LDB #100
LOOPI (PROCEDURE )
DECB
BNE LOOPI
```

"There is little difference in the amount of typing or entry time, so is there any other advantage of the assembler over Basic? I think so; first, during the procedure you can test for limits such as out of Memory, or Unstable conditions such as divide by zero, and exit out of the loop without any side effects. In the HLLs that I am familiar with, this can get you in deep trouble. Second, you may enter the Loop at any point, in effect using the same loop for several similar tasks. I don't recommend that for any HLL.

"A similar case can be made for the Decision structure, IF - THEN - ELSE in Basic. In assembler this would be written:

```
TST SWITCH (any variable, register, CC, etc.)
BNE PROC1 (THEN procedure)
PROC2 (ELSE procedure)
```

There are 16 different conditions that can be tested in assembler compared to 6 in Basic. In assembler you can test each BIT of the Byte, saving both memory and disk space."

#### RON'S NOTE

I can't wait to get this one in. Most HLLs, such as PL/9 and "C" allow such tests as:

```
IF BYTEVAR AND 7 = 7 THEN...
```

That would test for the low order three bits being ones. The point is that we can do the same tests in a HLL very simply. Now to return to Dan's comments.

"The most important advantage gained with the use of assembly language is the saving of memory and disk space. This has a snowball effect. If you have more room you can keep the whole file in memory. If the complete file is in memory you don't have to have a CHAIN structure, saving more memory; also, if the file is in memory it will run much faster and you don't have to output a message such as "PLEASE STAND BY, I AM PREPARING TO RUN YOUR PROGRAM" which saves even more memory.

"The file structure that I use permits both fixed and variable length fields. The fixed fields can store numbers as 2 digits per byte (packed). Variable length fields which are empty only require 1 byte. With this type of file structure my Payroll program can contain a year's records for 1500 employees on one 5" floppy disk (750K). I have seen a payroll program written in Cobol which could only keep 150 employees records on a 5 megabyte hard disk. I have been told that the Apple payroll program can only handle 50 employees per disk.

"Lately we have been seeing advertisements for programs which require a minimum of 128K of Ram and a Hard disk for operation. I can only conclude from this trend that many programmers are just moving programs from Minicomputers down to Micros and are just waiting for the micros to get faster and bigger so their programs will run at an acceptable speed. I have a full screen Editor that uses 8K of RAM, yet Wordstar takes 80K of Disk space and 20K of RAM."

# MY RESPONSE

I should mention that Dan preceded the above with "if you are going to agree with me, and not make some ridiculous statements, I don't see how we are going to have a big give and take discussion on programming." After the above quote he says "Chop me up in little pieces and I will get back at you in the next go-round." I will say that this little debate or what ever you choose to call it, has brought more reader comments than anything else I have ever written in this column.

First, let me once again indicate that the end use of the computer in my case may not be quite the same as in Dan's. My applications are generally in a stand-alone environment, controlling some mechanical device or measuring some input signal. I have no need for disk access since there is no disk drive in the end use of my programs. (I am speaking of my work in this comment).

My JUST program (see Southeast Media ad) is written in PL/9. It is slightly over 4K in length, and the listing is about 15 pages. It processes a line of text at a time, so that the RAM requirement other than for the program is trivial. So what if I could write it in Assembler and end up with a 2.5K to 3K program? It runs fast enough to keep up with most any printer that anyone would use for text.

The remainder of my rebuttal will boil down to two points; universality, and clarity or readability. Certainly the (procedure) in Dan's examples above will be a few lines to a few hundred lines of code. The procedure portion of the code in "C" or PL/9 could conceivably be a page of code. The higher level languages encourage making that procedure a separate "subroutine" or "function" or "PROCEDURE" (all equivalent in the various languages). Isn't the following clearer?

```
IF CHAR <$20
THEN DOCONTROL
ELSE PUTCHAR(CHAR); /* PL9 CODE */

for (i=1; i<=100; i++)
{
    procedure statements
}
```

The first example is PL/9 code. DOCONTROL would of course be a procedure somewhere above that code in the program to handle a control character.

The second example is a "C" for - next loop. The information in the parentheses includes from left to right, initial value for index, condition for remaining in the loop, and increment for index. "i++" indicates incrementing the loop by 1. "i+=3" would mean to increment by 3. Since "C" is a standard language, my loop written as above will run in a 68000 based system, a Z-80 based system, or for that matter any system for which a "C" compiler exists. If the disks aren't compatible, I can get my source text file from one system to the other via a Modem!

```
FOR N:=1 TO 100 DO
BEGIN
    (statements)
END;
```

There you have it in Pascal, another language that is available for almost any processor. That loop is therefore universal. Compare the "C" and Pascal versions. You will see that Pascal spells it out more readably, and "C" uses more of a shorthand notation. You have your choice.

I use PL/9 in my work, primarily because I am concerned more with efficient code generation than universality. PL/9 is efficient because it DOESN'T do a lot of internal checking. Define an array with a dimension of 100 and PL/9 reserves 100 locations. The index may vary from 0 to 99 (not 100). Now access array(200) and PL/9 won't complain, but your program will probably bomb. PL/9 is a lower level language than Pascal, and about on the same level as "C". Add 32767 to 32767 in either (using integer variables) and you will get -2. You will have generated a signed arithmetic overflow in the addition. Both C and PL/9 ignore overflow errors. The programmer has to know what he is doing!

You might think that is all bad. First of all, consider the fact that assembler code does no checking at all, unless you code the checking as part of your program. Secondly, if you have a FOR NEXT loop for VAR = 1 to 10, you hardly need to check VAR for overflow each time you increment it in the loop! There are many cases where checking is absolutely unnecessary. In the few where it is necessary, you can code it in as part of your program in PL/9 or "C". By the way, "C" doesn't care if you access an array with an out of range index either. In fact you can even use array(-3) in "C". If you know what you are doing, you might be able to take advantage of that to gain some efficiency. Don't try such tricks in Pascal, however. It will instantly tell you you made an error! It is a higher level language. The error checking costs some code, but is worth having if you need it, for example to check user input information. In my machine applications, the input is from an Analog to Digital converter and the range of values is completely predictable so error checking is unnecessary and would slow down the program operation needlessly.

I have a program that I was just looking at today. I wrote a machine program in Assembler some time ago. I decided to use 16 bit integer arithmetic and force a fixed decimal point three places from the right. That makes my minimum number 0.001, and my maximum 32.767. Of course I can have both positive and negative values in that range, and zero is a valid number. I did the whole program in 2K, and found that I was always exceeding the number range when I made any change in the program. I would always have to go back and re-scale things to make it work again. The listing is 25 pages.

I rewrote the program in PL/9 and it was reduced to 7 pages of highly readable program. I gained the use of a 7 digit floating point math package and scientific functions good to 7 places rather than my crude approximations in the assembler version that were good to 0.5% or so. I gained the possibility of formatted output of numbers through modifying one of the supplied library programs that allowed printing of numbers:

```
PRINTR(VALUE,6,2);
```

That would output the value of a floating point number in a field of 6 columns, with the two digits after the decimal point. I had to write a lengthy routine in assembler to do the same thing. The PL/9 compiler actually compiles the PL/9 program faster than the assembler assembles the assembler program, even though with all the floating point math package and added convenience features, the PL/9 program runs almost 6K. Note that I am not saying that the PL/9 compiler generates three times as much code as if I had used assembler. If I had to write a program with equal capabilities to the PL/9 program in assembler, I would first add my floating point math package and sci functions, about 2K, etc. I doubt that the PL/9 code would be more than 30% larger than the assembler code if the programs had equal performance.

Dan made one other point about multiple entry points in a subroutine in Assembler. I'll amplify on



his remark and explain what he meant. Let's go back to 6800 code. Suppose you want to increment the X register some variable number of times:

```
INCRX8 INX
INCRX7 INX
INCRX6 INX
INCRX5 INX
INCRX4 INX
INCRX3 INX
INCRX2 INX
INCRX1 INX
RTS
```

Now if we want to increment X by 4 we JSR INCRX4. If we want to increment it by 8 we JSR INCRX8. That is a very clever and efficient way to do the increment with a limited instruction set like the 6800 one. In 6809 code we wouldn't need a subroutine at all.

```
LDB #4
ABX
```

or

```
LEAX 8,X
```

These would do the same job. This is a simple example of a multiple entry point subroutine. In PL/9 of course you wouldn't be incrementing a processor register directly, but perhaps an analogy would be a variable delay time procedure:

```
DELAY (BYTE COUNT);
  WHILE COUNT >0 COUNT = COUNT - 1;
ENDPROC;
```

To call the delay procedure in a program we would use:

```
DELAY(10);
or
DELAY(255);
```

Dan describes his file structures. The descriptions essentially fit FLEX Sequential files and Random Access Files. Sequential files have variable length fields. Random Access Files have fixed length fields.

In one area, I can't disagree. Dan mentions the ads for programs that require 128K of RAM and a Hard disk for operation. There is no doubt that some of the early compilers for Microprocessor based computers WERE very inefficient. Some of those are still around. The point is that compilers are not all alike. I'm not advocating using just anything that happens to be laying around, but I am saying that GOOD, EFFICIENT, FAST compilers do exist in a variety of languages, and that no one is going to convince me that he can write 30 pages of assembler code and debug it as fast as I can write 7 or 8 pages of compiler source code and debug my program. I know that I can get my program written and debugged three or four times as fast as the assembler programmer doing the equivalent program. If a bug does show up for a particular condition, I can generally pinpoint it and have it repaired in a few minutes.

Well, I think I have said enough. I'll let Dan have the last word, since I started this debate. Dan, go ahead and reply to this and I'll get it in a column in a couple months.

## SUPPORT YOUR ADVERTISERS

# OS9 USER NOTES

By: Peter Dibble  
517 Goler House  
Rochester, NY 14620

## Big System Hardware

Gimix has offered CoCo owners an attractive deal. Gimix will accept a CoCo in trade for a Gimix system at 110 percent of its value. Even with this roughly thousand dollar break in the price of a Gimix the upgrade is expensive, but, speaking as a person who has used a Gimix for many many hours, if you can find the money, take this opportunity. What makes it worth thousands of dollars to move from a CoCo to a SS50 system? The most important difference is that everything works right on the larger systems. Another is that the more expensive systems are faster. A two megahertz 6809 runs more than twice as fast as a CoCo in its normal mode. The DMA disk controller and other powerful I/O devices also make a noticeable difference.

The upgrade from a CoCo to a SS50 system isn't the end of the line. All the major SS50 systems that support OS-9 support both OS-9 Level One and Level Two. The move to Level Two involves a new version of the OS-9 operating system, but no change in applications programs. All the modern SS50 systems I know of can be upgraded with little or no change to the hardware (the main requirement is memory management hardware). I imagine that OS-9 Level Two might run with the 56K of memory that Level One uses, but just barely. Level Two begins to come into its own at 128K. At 344K, I have never run out of memory.

## Big System Software

There is a bit of controversy arising in the OS-9 world. Smoke Signal Broadcasting has been responsible for a lot of 6809 software over the years. There is even an operating system which they are responsible for. Now they are contributing to OS-9 software. My understanding is that Smoke commissioned someone to work on the version of OS-9 licensed to them. Their consultant made OS-9 less modular in order to improve its performance. The Smoke users I know confirm that the revisions make the Smoke version of OS-9 run faster than it used to. Running faster would seem to be an advantage, but the changes Smoke has made turn out to be a mixed blessing. There appear to be subtle incompatibilities between OS-9 as it comes from Microware and OS-9 from Smoke Signal Broadcasting. I have spoken to Microware and they say that they can't support Smoke's version of OS-9 (that may have changed by the time you read this). I have had trouble exchanging software with Smoke users.

The Smoke users are amazingly tolerant. I have read exchanges on the Compuserve OS-9 SIG in which Smoke users exchange tips on ways to prevent the DIR command from intermittently producing junk.

I certainly approve of improving OS-9's performance, but it is very important that an operating system be as standard as possible. If I were buying a system from Smoke Signal Broadcasting, I would want strong assurances that their version of OS-9 was compatible with Microware's on every level. A good test would be that all applications programs and system modules that run under standard OS-9 should run under the modified one, and vice versa.

# Plan to Attend the 3<sup>rd</sup> ANNUAL OS-9 USERS SEMINAR

**August 17, 18, 19, 20  
Pre-Registration Only!**

## • MORE INFORMATION

- MORE EXHIBITS
- MORE SPEAKERS
- HARDWARE
- SOFTWARE



## • TECHNICAL SESSIONS FOR 6809 & 68K

Plan now to attend the 3rd Annual OS-9 User Seminar. This is an event you won't want to miss if you use, sell or are interested in systems that use Microware 6809/68000 software. Informative round-table discussions on almost every aspect of the design and use of Microware software will be held. A bigger and better exhibit area will have display booths from many of the leading suppliers of OS-9 compatible hardware and software. Don't miss this chance to increase your knowledge and skill in the latest microcomputer software technology — Register today!!!

**Fee: \$125**

**Location: Marriott Hotel  
Des Moines, Iowa**

**Don't Miss It: Pre-Register Now!  
Call: 515/279-8844 or Write:**

Boy, everyone is having a great time . . . You bet, Jeanne.



## The Compuserve OS-9 SIG

The OS-9 Special Interest Group on Compuserve is booming. Messages flow through the bulletin board so fast I am beginning to question my ability to read them all. Many experienced OS-9 users regularly check in, but it is a particularly good resource for newcomers. I strongly suggest that you join Compuserve if there is an access point close to you. It is worth it even if you only use it to access the OS-9 SIG.

## OS-9 on the Color Computer

I have been saying nasty things about Tandy which aren't true. I blamed the sloppy programming in the CDDISK device driver on Tandy when it seems the blame should fall on Microware and Microsoft. The bootstrap for the CoCo is in ROM. There is only one bootstrap ROM, designed by Microsoft for use with Color Disk Basic (I guess). Microware had to design the CoCo implementation of OS-9 so it could be loaded with that Bootstrap. The CoCo boot ROM reads 15 sectors off track 35 into a fixed location in memory. The OS9Boot file had to fit into those 15 sectors. This memory constraint forced Microware to pay even more attention to writing compact code than they usually do. Since 6809 instructions that do direct memory references take less memory than indexed instructions, Microware used them whenever they could. Since versatile device drivers take more memory than limited drivers, they wrote limited drivers. Tandy, I apologize for the nasty thoughts I sent your way.

I decided to write this month's project for the CoCo. I noticed that Color Basic has a number of commands which make assorted honks and beeps emerge from my TV. Basic09 has no way to make those noises. I checked the "Color Computer Technical Reference Manual" for information about the sound generator, and found that the Color Computer generates sound with a Digital to Analog converter. The output from the D/A converter is routed through an analog multiplexer to the modulator, and hence to the TV. It looked like OS-9 could learn to make noise.

I expect that the reason Microware didn't include sound generation in their OS-9 for the Color Computer is that sound generation with an D/A converter is a very time dependent operation. A note is played by gradually (in computer terms) raising and lowering the voltage generated by the D/A converter. This has to be done with a timing loop in a program. The timing loop must have exclusive use of the computer, or the rate at which the voltage rises and falls will vary causing the note being generated to rise and fall. Some people might find the resulting yodel surprising. A program can give itself exclusive use of the computer by masking out interrupts, but locking out interrupts for more than a few millionths of a second is antisocial behavior for any program -- even a part of the operating system.

Still, the ability to at least be able to generate a beep seems important to me. I started by writing a program called Sound to investigate sound production. The program generates a saw-tooth wave that sounds rather like a saber saw cutting thin plywood, but it works. The most important discoveries I made while writing Sound were how to initialize the multiplexer so the D/A converter's output would be routed to the TV. The control registers at \$FF03 and \$FF23 both need to be modified. The fact that they could be modified was another interesting discovery. I am used to control registers being either readable or writeable. These registers are to some extent read/write. CoCo

programmers may take this for granted, but I was pleasantly surprised.

Once the control registers are set, sound can be generated by simply writing different values into the most significant 6 bits of the byte at \$FF20. The faster the value is changed the higher the pitch. I wrote the program to send 1000 waves, then stop.

There is lots of room for improvement in sound. The quality of the note created by the program could be improved, and the program might even be made to play a song. I decided to drop Sound and work on building a Device Driver for the D/A converter.

The Device descriptor I wrote for the D/A converter, Beep, is almost as small as a Device Descriptor can be. The D/A converter is not a random access device so I decide to use the SCF file manager to drive it. There are no options except the one byte which indicates that it is a SCF device. There are three addresses in the descriptor. Normally a descriptor only needs one port address, but in this case, since the three addresses used in making the D/A converter make sound aren't related, I included all the addresses explicitly.

The Device Driver, called Beeper, is not interrupt driven. Most OS-9 device drivers use interrupts to give them a way to avoid wait loops, but I couldn't find a way to get the D/A converter to generate interrupts. In this case interrupts weren't necessary; the device responds as fast as data can be pumped into it.

The Initialization entry puts some values that will be needed in the termination routine into device static storage, and sets the two PIA registers that need to be adjusted to permit sound to be made. The termination entry sets the two control registers back the way they were before Beeper started, and the GetStat and PutStat entries don't do anything at all. The read and write entries deal with the fact that the D/A converter only uses the high-order six bits of the register it is accessed through.

#### Installation of Beep/Beeper

Beep and Beeper have to be typed in and assembled. As usual, the USE statements between the IFP1 and ENOC don't come out in the assembly listing. You will have to include use statements for both OS9DEFS and SCFDEFS for these programs. When you assemble the Beeper file it will generate a file in the execution directory called Beeper with both Beep and Beeper in it.

To use beeper first load it with the OS-9 command line:

OS9: load beeper then link beeper with the command line:

OS9: link beeper Since beeper is the second module in the file it will have a tendency to disappear if you don't link it.

As a first try you can get a low growl out of your computer by listing a file to /Beep. I used OS9: list beeper >/beep

To get a more interesting sound out of the device you will need to feed it meaningful data. The Basic09 program called TestBeep generates a thousand bytes of sin wave. TestBeep is intended to be pecked and run out of the execution directory. If it is run from source the BYE should be removed. It takes a long time to initialize the array, so be

patient. The wave can be sent one byte at a time with a loop like:

```
for i=1 to 1000
  put 0ound,note(i)
next i
```

But OS-9 doesn't do very well at outputting a single character at a time. This program segment demonstrates that by generating a low, raspy note. To get a higher, smoother note I sent the entire thousand-byte array with one write. The quality of the tone still leaves a lot to be desired, but it's the best I could do quickly.

#### Applications for /Beep

I imagine that the timbre of the tone generated by TestBeep could be improved by spending more time with the wave form: the rough sin wave I use is pretty crude. Certainly the pitch can be varied by changing the frequency of the wave. I discovered that TestBeep just as it stands is a useful demonstration of OS-9's multitasking behavior. I started TestBeep with the command line:

OS9: BASIC09 TestBeep& if you have RUNB

OS9: TestBeep& will work fine. This runs the program as a background task. When the noise started, I ran a variety of different programs and noticed the effect on the sound.

If you want to generate a higher pitch than you can get out of Beeper, I suggest doing more work in the device driver. The approach I have in mind is to add a buffer in the device static storage for Beeper. When Beeper receives a request to write a zero value it will load the next 256 bytes written into the buffer. When the buffer isn't being loaded, each value written to Beep will indicate a number of times to send the buffer out the D/A. I believe that this approach will prove to be really useful, especially if there is a default wave loaded into the buffer by the INIT code.

#### The Users Group

I hope all the members of the OS-9 Users group will have their disks by the time you read this. I am afraid that some of you will have received the wrong type of disk. I am responsible for this. We don't have any record of the type of disk (size and format) any of our early members use. Some of the people who have joined recently have included information about their disk, but in most cases I have had to guess. If you get a disk you can't deal with, write to the Users Group address, and we will try to get you a disk you can read.

00001		nam	Sound
00002		ttl	Sound generator for CoCo
00003		IFP1	
00005		ENOC	
00006	0011	TYPE	SET PRGM:OBJECT
00007	0000 07CD00065	MOD	EXDSND,NAM,TYPE,REENT+1,ENTRY,0512E
00008	0 0000	RND	2 Address of B/A control register
00009	0 0002	RND	2 Address of another B/A control
00010	0 0004	RND	2 Address of B/A input
00011	0 0006	RND	2 Number of waves to send
00012	0 0008	RND	1 Initial value of first Control
00013	0 0009	RND	1 Initial value of other control
00014	0 000A	RND	200 STACK
00015	0 00B2	OS12E	EDU
00016	0000 534F554E	NAM	FCS /SOUND/
00017	0012	ENTRY	EDU
00018	*****		
00019	0		Initialize addresses in local storage
00020			
00021	0012 CCF723	LDD	00FF23
00022	0015 DB00	STD	CNTL
00023	0017 CCF720	LDD	00FF20
00024	001A 3B04	STD	PORT



```

00025 001C CFFF03      LOD  BFFF03
00026 001F 0002      STB  CNTL2
00027      *
00028      * Save initial values of control registers
00029      * and set them to route B/A output to sound
00030      *
00031 0021 A6D4      LDA  [CNTL,U]
00032 0023 9708      STA  CNTLV
00033 0025 8A00      DRA  #800
00034 0027 A7D4      STA  [CNTL,U]
00035 0029 A6D002     LDA  [CNTL2,U]
00036 002C 9709      STA  CNTL2V
00037 002E 84F7      ANDA  #8FF-89B
00038 0030 A7D002     STA  [CNTL2,U]
00039      *
00040      * Initialize the counter
00041      *
00042 0033 C03E8      LOD  #1000
00043 0036 D006      STD  CNTR
00044      *
00045      * Send waves
00046      *
00047 0038      LOOP2
00048 003B 8600      LDA  #0
00049      *
00050      * Send each wave
00051      *
00052 003A      LOOP1
00053 003A A7D804      STA  [PORT,U]
00054 003D 8B04      ADDA  #4
00055 003F 12      NOP
00056 0040 12      NOP
00057 0041 12      NOP
00058 0042 12      NOP
00059 0043 12      NOP
00060 0044 12      NOP
00061 0045 12      NOP
00062 0046 12      NOP
00063 0047 12      NOP
00064 0048 8100      CNPA  #0
00065 004A 26EE      BNE  LOOP1
00066      *
00067      * End of sending one wave.
00068      * See if we still need to send more
00069      *
00070 004C DC06      LDB  CNTR
00071 004E 830001     SUBD  #1
00072 0051 D006      STD  CNTR
00073 0053 26E3      BNE  LOOP2
00074      *
00075      * Restore initial values to control registers
00076      *
00077 0055 9608      LDA  CNTLV
00078 0057 A7D4      STA  [CNTL,U]
00079 0059 9609      LDA  CNTL2V
00080 005B A7D002     STA  [CNTL2,U]
00081 005E 5F      CLRB      clear carry
00082 005F 103F06     OSR  F8E11T  return to DS-9
00083 0062 528D69     ENDD
00084 0063      ENDSND EQU  *

```

```

00000 error(s)
00000 warning(s)
00045 00101 program bytes generated
00082 00210 data bytes allocated
00EF8 03832 bytes used for symbols

```

```

TTL DEVICE DESCRIPTOR
NAM BEEP
TYPE SET DEVIC+OBJECT
MOD BPEND,BPRNAM,TYPE,REENT+1,FNAME,DRVNAM
FCB READ,+WRITE, MODES
FCB #FF,#FF,#20 PORT ADDRESS

FCB OPTL Length of options section
OPTIONS MOD *
FCB DT.SCF
OPTL EQU *-OPTIONS

```

```

CNTL1 FDB #FF23 address of control byte 1
CNTL2 FDB #FF03 address of control byte 2

```

```

BPRNAM FCS /BEEP/ name of this module
FNAME FCS /SCF/ File Manager name
DRVNAM FCS /BEEPER/ Device driver name
ENDD
BPEND EQU *

```

```

NAM BEEPER
IFP1
USE /MD/DEFS/DS9DEFS.1
USE /MD/DEFS/DS9LDEFS.1
USE /MD/DEFS/DS9SCFDEFS.1
ENDD
USE BEEP Device Descriptor
TTL DEVICE DRIVER FOR D/A
PAGE
TYPE SET DRVIR+OBJECT
REVS SET REENT+1
MOD BPEND,BPRNAM,TYPE,REVS,ENTER,REMSIZE
FCB READ,+WRITE, DRIVER MODE
BPRNAM FCS /BEEPER/
FCB #1 EDITION
*****
* Device Static storage
*
ORG V.SCF System part of Static Storage
*****
* Local part of static storage*
*
PORTA RMB 2 PORT ADDRESS
CTL1V RMB 1 HOLD CNTL1 VAL
CTL2V RMB 1 HOLD CNTL2 VALUE
CTL1A RMB 2 HOLD CNTL1 ADDR
CTL2A RMB 2 HOLD CNTL2 ADDR
REMSIZE EQU .

```

```

*****
* Entry vectors
*
ENTER-
LDBA INIT
LDBA READ
LDBA WRITE
LDBA GETSTAT
LDBA PUTSTAT
LDBA TERM
spc 1
INIT
*****
* U ADDRESS OF DEVICE STATIC STORAGE
* Y ADDRESS OF DEVICE DESCRIPTOR MODULE
*
LDI CNTL1,Y Get control address 1 out of Descriptor
STX CTL1A,U Save the address
LDA ,X Get the present value of cntl1
STA CTL1V,U save it for later restore
ORA #80B set it for sound
STA ,X

LDI CNTL2,Y do the same stuff for cntl2
STX CTL2A,U
LDA ,X
STA CTL2V,U
ANDA #8FF-89B
STA ,X

CLRB CLEAR CARRY
RTS RETURN
spc 1
READ
*****
* U ADDRESS OF DEVICE STATIC STORAGE
* Y ADDRESS OF PATH DESCRIPTOR
* RETURN CHARACTER READ IN A
*

```

```

LDI V.PORT,U port address from device descriptor
LDA ,I D/A value
LSRA
LSRA Shift out low order bits
CLRB Clear carry
RTS

```

```

spc 1

```

```

WRITE

```

```

*****

```

```

* U DEVICE STATIC STORAGE

```

```

* Y PATH DESCRIPTION

```

```

* A VALUE TO WRITE

```

```

*

```

```

LDI V.PORT,U

```

```

LSLA Shift out high order bits

```

```

LSLA

```

```

PSHS A save value to write

```

```

LDA ,I Get current value at Port

```

```

ANDR #20000011 clear D/A value

```

```

ORA ,S+ put value to write in

```

```

STA ,I send it

```

```

CLAB

```

```

RTS RETURN

```

```

spc 1

```

```

GETSTAT

```

```

PUTSTAT

```

```

CLRB

```

```

RTS

```

```

TERM

```

```

*****

```

```

* U DEVICE STATIC STORAGE

```

```

*

```

```

LDI CTL1A,U restore original Cntl1 value

```

```

LDA CTL1V,U

```

```

STA ,I

```

```

LDI CTL2A,U restore original Cntl2 value

```

```

LDA CTL2V,U

```

```

STA ,I

```

```

CLRB clear carry

```

```

RTS

```

```

ENDR

```

```

***** EQU *

```

```

00001      NAM BEEPER
00002      (FP)
00006      ENDC
00007      USE DEEP Device Descriptor
00008      TTL DEVICE DESCRIPTOR
00009      NAM DEEP
00010      00F1 TYPE SET DEVIC+OBJECT
00011      0000 87C00027 MOD BPEXD,BPMAN,TYPE,REENT+1,FILENAME,BPMAN
00012      0000 03 FCB READ,+WRITE. NODES
00013      0000 FFFF20 FCB $FF,$FF,$20 PORT ADDRESS
00014
00015      0011 01 FCB OPTL Length of options section
00016      0012 EQU *
00017      0012 00 FCB DT.SCF
00018      0001 OPTL EQU *-OPTIONS
00019
00020      0013 FF23 CNTL1 FDB 0FF23 address of control byte 1
00021      0015 FF03 CNTL2 FDB 0FF03 address of control byte 2
00022      0017 42454500 BPMAN FCS /DEEP/ name of this module
00023      0018 5343C6 FILENAME FCS /SCF/ File Manager name
00024      001E 42454530 BPMAN FCS /BEEPER/ Device driver name
00025      0024 5BAE13 ENOD EQU *
00026      0027 BPEXD EQU *
00027      TTL DEVICE DRIVER FOR D/A
00028      00E1 TYPE SET DRIVER+OBJECT
00029      0001 REVS SET REENT+1
00030      0000 87C00076 MOD BPEXD,BPMAN,TYPE,REVS,ENTER,NEWSIZE
00031      0003 03 FCB READ,+WRITE. DRIVER MODE
00032      000E 42454530 BPMAN FCS /BEEPER/
00033      0014 01 FCB 1 EDITION
00034      *****
00035      * Device Static storage
00036      *
00037      0 0010 ORG V.SCF System part of Static Storage
00038      *****
00039      * Local part of static storage
00040      *

```

```

00041      0 0010 PORTA RMB 2 PORT ADDRESS
00042      0 001F CTL1V RMB 1 HOLD CNTL1 VAL
00043      0 0020 CTL2V RMB 1 HOLD CNTL2 VALUE
00044      0 0021 CTL1A RMB 2 HOLD CNTL1 ADDR
00045      0 0023 CTL2A RMB 2 HOLD CNTL2 ADDR
00046      0 0025 NEWSIZE EQU *
00047

```

```

00048      *****

```

```

00049      * Entry vectors

```

```

00050      *

```

```

00051      0015 ENTER

```

```

00052      0015 16000F LBRM INIT

```

```

00053      0018 16002C LBRM READ

```

```

00054      0018 160031 LBRM WRITE

```

```

00055      001E 16003E LBRM GETSTAT

```

```

00056      0021 160038 LBRM PUTSTAT

```

```

00057      0024 16003A LBRM TERM

```

```

00058      0027 INIT

```

```

00059      *****

```

```

00060      * U ADDRESS OF DEVICE STATIC STORAGE

```

```

00061      * Y ADDRESS OF DEVICE DESCRIPTOR MODULE

```

```

00062      *

```

```

00063      0027 AEAB13 LDI CNTL1,Y Get control address 1 out of D

```

```

00064      002A AFCB21 STL CTL1A,U Save the address

```

```

00065      0020 A6B4 LBRM ,I Get the present value of cnt1

```

```

00066      002F A7CB1F STA CTL1V,U save it for later restore

```

```

00067      0032 BA0B ORA #0B set it for sound

```

```

00068      0034 A7B4 STA ,I

```

```

00069

```

```

00070      0036 AEAB13 LDI CNTL2,Y do the same stuff for cntl2

```

```

00071      0039 AFCB23 STL CTL2A,U

```

```

00072      003C A6B4 LBRM ,I

```

```

00073      003E A7CB20 STA CTL2V,U

```

```

00074      0041 B4F7 ANDR #0FF-60B

```

```

00075      0043 A7B4 STA ,I

```

```

00076

```

```

00077      0045 5F CLRB Clean carry

```

```

00078      0046 39 RTS RETURN

```

```

00079      0047 READ

```

```

00080      *****

```

```

00081      * U ADDRESS OF DEVICE STATIC STORAGE

```

```

00082      * Y ADDRESS OF PATH DESCRIPTION

```

```

00083      * RETURN CHARACTER READ IN A

```

```

00084      *

```

```

00085      0047 AE41 LDI V.PORT,U port address from device descr

```

```

00086      0049 A6B4 LBRM ,I D/A value

```

```

00087      004B 44 LSRA

```

```

00088      004C 44 LSRA

```

```

00089      004D 5F CLRB Shift out low order bits

```

```

00090      004E 39 RTS Clear carry

```

```

00091      004F WRITE

```

```

00092      *****

```

```

00093      * U DEVICE STATIC STORAGE

```

```

00094      * Y PATH DESCRIPTION

```

```

00095      * A VALUE TO WRITE

```

```

00096      *

```

```

00097      004F AE41 LDI V.PORT,U

```

```

00098      0051 48 LSRA Shift out high order bits

```

```

00099      0052 48 LSRA

```

```

00100      0053 3402 PSHS A save value to write

```

```

00101      0055 A6B4 LBRM ,I Get current value at Port

```

```

00102      0057 B403 ANDR #1000000011 clear D/A value

```

```

00103      0059 AAE0 ORA ,S+ put value to write in

```

```

00104      005B A7B4 STA ,I send it

```

```

00105      005D 5F CLRB

```

```

00106      005E 39 RTS RETURN

```

```

00107      005F GETSTAT

```

```

00108      005F PUTSTAT

```

```

00109      005F 3F CLRB

```

```

00110      0060 39 RTS

```

```

00111      0061 TERM

```

```

00112      *****

```

```

00113      * U DEVICE STATIC STORAGE

```

```

00114      *

```

```

00115      0061 AECB21 LDI CTL1A,U restore original Cntl1 value

```

```

00116      0064 AACB1F LBRM CTL1V,U

```

```

00117      0067 A7B4 STA ,I

```

```

0011B 0069 AECB23      LDZ  CTL2H,U  restore original Ctl2 value
00119 006C A6CB20      LDR  CTL2W,U
00120 006F A7B4        STA  ,I
00121 0071 5F          CLRB          clear carry
00122 0072 39          RTS
00123 0073 A1B2B1      ENDB
00124 0076            BPREMB  EDU  +

```

```

00000 error(s)
00006 warning(s)
0009D 00157 program bytes generated
0000B 0000B data bytes allocated
0164B 05707 bytes used for symbols

```

```

PROCEDURE TESTBEEP
0000  DIM NOTE(1000):BYTE
000C  DIM I:INTEGER
0013  DIM SOUND:INTEGER
001A  OPEN "SOUND","BEEP":WRITE
002A  FOR I=1 TO 1000
003B  NOTE(I)=32+11*SIN(111
0052  NEXT I
005D  FOR I=1 TO 100
006B  PUT "SOUND,NOTE
0077  NEXT I
0082  BYE

```

## "C" User Notes

Norm Conno  
3 Pryor Road  
Natick, MA 01760

Ahhhh. Finally. Some time to sit down at the keyboard and get out a column before you folks start to think I've quit. It's been really hectic the past few months. My schedule just doesn't allow enough time to both develop programs and write columns. Since programs run computers and not columns, I've spent the last month getting the modem program going.

I've also run across a couple of bugs in the Windrush compiler and have received the new Introl version 1.5 compiler. I'll start with my Windrush problems, touch on the new Introl compiler and finally get around to the modem program.

### WINDRUSH BUGS

The first bug I found was that static float initialization was broken; the second "bug" was that their implementation of `getchar()` and `gets()` left a lot to be desired. You probably wouldn't have run across the first bug unless you're into more advanced programming; but the second bug has probably caused you problems if you're doing any serious work with the compiler at all.

These bugs, along with a couple of other things that turned out to be different interpretations of the standard than I've previously seen, were duly reported to Windrush and James McCosh.

Well, it took a few weeks, but yesterday I got a letter from James which contained an explanation of his interpretation of some implementation details and a promise that the real bugs would be fixed.

I won't bore you with all the details, but it essentially boils down to the fact that developmental work on the compiler is done under Uniflex and then transported over to FLEX. Since Uniflex and FLEX present vastly different programming environments, some bugs crept in.

So it looks like Windrush is continuing to provide the level of support that they have in the past and that we have come to expect of them.

### INTROL VERSION 1.5

Introl is now shipping version 1.5 of their compiler. For all practical purposes, this is a completely new package. In talks with Rich Pennington, chief architect of the package, I was told that something like 20,000 new lines of code went into the package. This translates into improvements in the compiler in terms of code speed and density as well as new compiler options and brand new support programs such as the assembler, linker and librarian.

The old assembler has been replaced with a real relocating assembler. The linker has been updated and split into two sections; a linker and a loader. And there is a new librarian with a lot more features than the old one.

The documentation is really well done with a quick intro and overview to get you started, followed by more detailed sections on each aspect of the package.

I have not had many opportunities to play with it. What little that I have used seems to go ok. There was one minor hitch with the `#include` directive. I used

```
#include "O.STDIO.H"
```

but it couldn't seem to find the file, so I got around it by copying `STDIO.H` onto the working disk and using

```
#include "STDIO.H"
```

As of this writing, I have not called them about it so it might even be fixed by now.

The modem program and some of the other support programs that will be published in this series were all compiled using this compiler.

No rating yet since I want to wait until I've done more testing and have had a chance to get some inputs from other Introl users that I know. A more complete review will come in a later column.

### MODEM9

After much dawdling and many interruptions, MODEM9 is done. This is a program based on the XMODEM protocol developed by Ward Christlanson which is now in the public domain. It is a nice protocol since data is transferred in packets with retries if errors are encountered.

This particular protocol was picked for MODEM9 because it is probably the one of the most widely used; by virtue of being the standard for RCPM (Remote CPM) bulletin boards. Why would we want to use it in the 6809 community? The first reason is that it seems to work well and I am unaware of any other widely accepted standard within our camp. However, having made that statement, I'll probably get a lot of mail telling me that I'm all wrong. Secondly, it allows us to log into the many (!) RCPM's around the country and pick up a lot of programs in source form; which we can then adapt to our systems.

There is also a growing number of "cnodes", which are bulletin boards dedicated (for the most part) to the C language and containing many interesting programs. What is available? I checked into the nearest cnode in my area. It had 24 "libraries" of C programs. I scanned a few of these libraries and here is a small portion of what I found.

games	from action oriented to adventures
ed	the Small-C editor from Dr. Dobbs
xlisp	a lisp like object oriented language
roff	a text processor similar to TSC's
utilites	galore

On most RCPM's you will find a mix of languages such as 8080/280 assembler, Pascal and BASIC. We may not be able to use it directly, but much of it can be adapted so why shouldn't we join the fun.



## XMODEM PROTOCOL

The xmodem protocol is based on sending 128 byte data packets. Each data packet is preceded by a header section and trailed by a simple longitudinal checksum. If a packet is received without error, it is acknowledged and the next packet is sent; otherwise the same packet is resent.

The packets size of 128 bytes was (presumably) picked because it was the size of a CP/M disk sector. The header section consists of three bytes, Ascii SOH, the sector number and its one's complement. This is shown pictorially below:

0	SOH (0x01)
1	sector number
2	sector number
3	data
4	data
5	data
.	....
.	....
.	....
130	data
131	checksum

Sector numbers start at 1, but rollover to 0 on the 256th sector. All data, the sector number and the checksum are sent as 8 bit data with no parity and one stop bit. The ACIAs (or UARTS) at both ends must be initialized accordingly. No restrictions are put on the actual data. It can be code or text. The handshaking codes used by the protocol are Ascii control codes and have the following meanings:

SOH	0x01	start of sector
EOT	0x04	no more data (end of file)
ACK	0x05	no error in sector
NAK	0x15	error in sector
CAN	0x18	stop the transfer

File transfers are essentially "receiver" driven. That is, the computer receiving the file controls information flow. The sending program waits for about a minute, or until it gets an initial NAK from the receiver. It then sends a packet and waits for the response. If the response is ACK, then it sends the next sector. If the response is NAK, or any other spurious response, then the same packet is resent. The sender finally sends an EOT to signify that no more sectors are available.

Here is a typical dialog between a sender and a receiver, including some errors, that shows how the protocol is driven by the receiver. It is excerpted from a bulletin board file authored by Ward Christianson.

sender	receiver
waits for one minute or until NAK received	
	timeout after 10 sec.
	<--- NAK
packet #1	---->
	<--- ACK

packet #2	----> (data byte gets hit)
	<--- NAK
packet #2 (retry)	---->
	<--- ACK
packet #3	---->
(ack gets hit)	<--- ACK
packet #3 (retry)	---->
	<--- ACK
EOT	---->
	<--- ACK

It's really the one minute wait by the sender that allows the receiver to timeout and send the initial NAK making the protocol effectively receiver driven.

## HARDWARE REQUIREMENTS

MODEM9 was designed to work with any terminal. Except for the statement that clears the screen, which is for an M19, it expects nothing more than a tty. The ACIA used for the modem is expected to be a 6850 with both its transmit and receive clocks running at 16 times 1200 baud. This allows the program to change the baud rate from 1200 to 300 baud. No interrupts are used. The program is used here regularly at 1200 baud with no trouble.

The program contains its own printer driver routines for a 6821 PIA based parallel port.

No modem control functions such as autodial are built in. It assumes that you will do all of that sort of stuff by hand.

Customizing it for your system should involve only changing the clear screen (or deleting it) and setting the #define's for your port addresses. If you use a serial printer, you might even be able to use the terminal drivers for the printer.

## PROGRAM ORGANIZATION AND USE

The program is composed of 4 modules. These are:

modem.h	header file
modem1.c	main program and terminal mode
modem2.c	xmodem send/receive functions
modem3.c	hardware drivers

The program has been compiled, essentially unchanged, with the Intel V1.4, Intel V1.5 and Windrush compilers. If you are using the Windrush compiler you will have to take all the global variables out of modem.h and properly declare them in the other files.

The program comes up in the command mode. From the command mode you can enter the other modes or change various program parameters. A brief description of the commands follows. Commands that need more input prompt the user for it.

- B aud
  - Change the baud rate of the modem port.
- C capture
  - Toggle the capture buffer on and off.
- D download
  - Receive a file from the host computer using the XMODEM protocol.
- E erase
  - Erase the capture buffer.
- F file
  - Set up what kind of file will be sent/received using the XMODEM protocol.

- H**elp
  - List out all the commands to the console.
- K**eep
  - Save (keep) the capture buffer in a file and then erase it.
- L**ist
  - List out the capture buffer to the console.
- P**rint
  - Print the capture buffer to the parallel printer.
- Q**uit
  - Return to FLEX.
- S**ettings
  - List out all the present parameter settings.
- T**erminal
  - Enter the terminal emulator mode. To escape hit hit NULL, which is control-shift-2.
- U**pload
  - Send a file to the host computer using the XMODEM protocol. Upload will prompt you for a file name.
- V**iew
  - Turn on the viewing option to see incoming or outgoing data when using the XMODEM protocol.

More needs to be said about the terminal mode. In most of the versions that I have run across, commands are issued while in the terminal mode. This necessitates using control codes for the commands. I felt that this was bad. Many editors available on big systems use control codes as part of their command set. If you use MODEM9 to log into your work account, having the MODEM9 commands as control codes could impact the usefulness of your editor. Those of you who use EMACS will appreciate what I mean.

This implementation of the terminal mode uses the NULL code to escape back to the command mode. Because the NULL code is rarely used for anything other than padding it's a pretty safe code to use. All other codes go to the host unfiltered.

#### ACCESSING HOST COMPUTERS

MODEM9 is designed to be used interactively. A typical session would be to get into MODEM9 and then dial up the system you want to talk to. When you get the carrier and have set the modem from talk to data, then enter the terminal mode. We will assume that you have logged into an RCPM with the XMODEM program on it. Perusing through the directory of the RCPM, you see a file that you would like. To download the program to your system you must do the following four steps.

- 1) Invoke the host's transfer program, calling out the send mode and the desired filename. That might be something like:

```
"XMODEM -S FILENAME"
```

- 2) Exit the terminal mode back to the command mode. Then enter the D command.
- 3) Enter the filename to store the data in.
- 4) Return to the terminal mode when the transfer is over.

Uploading a file is done in a similar fashion. I should point out that most of the bulletin boards give you the info on how to use their XMODEM or its equivalent as part of the login procedure.

This may seem like a lot to do quickly, but don't forget, you've got about a minute to get steps 1 to 3 done.

MODEM9 has a number of error modes that it checks for. Besides checksums, there are timeouts, missing packets and a few others. If a transfer does get clobbered to the point where XMODEM9 aborts then the only thing you can do is to get back to the terminal mode and fire back a few control-x characters. Control-x is the cancel code. Technically, it's not part of the Christanson spec. However, I've read where many of the RCPM's put it in. I have only had this happen to me once.

All though I have not tried it, you should be able to use this program to exchange software with other 6809 users. To do this will require some cooperation though. After phoning the desired party, both would enter the terminal mode. This would allow them to "talk" to each other through their terminals. (Note that each user must hit both <cr> and <lf> at the end of a line) To transfer a file, one enters the download mode while the other enters the upload mode. You can even exchange binary files this way. I think that you will enjoy using this program. I do.

#### CAVEATS

There are a couple of remaining rough spots and caveats to be aware of. Sometimes when I log off my work system and the host modem drops carrier the modem port will hang and hitting NULL won't take the program out of the terminal mode. I haven't investigated it.

There are three file modes; Ascii, Binary and CP/M. What these modes do is change how the FLEX file gets opened. Here is a short synopsis. Remember that when you're sending a file it's opened for read at your end.

#### Ascii

- Read:** Open file in the text mode. Send all characters as they are.
- Write:** Open file in the text mode. Tabs are expanded to spaces before being written to the file. All other controls codes except <cr> are filtered out.

#### Binary

- Read:**
- Write:** Open file in the binary mode and send all characters as they are.

#### CP/M

- Read:** Open file in the text mode. Add a <lf> after each <cr>.
- Write:** Open file in the binary mode and send all characters as they are.

Sending a file to a CP/M system in the CP/M mode has been done in both directions on my 8088 system. Since CP/M terminates all lines with <cr><lf> and allows tabs (which it assumes are 8 spaces) it's easier to just receive them in the binary mode and then post-process them. The programs RLIST and BCOPY do just that. They open read files as binary and then expand tabs and swallow any control characters other than <cr>.

#### USING THE CAPTURE BUFFER

You might be wondering what the capture buffer is used for. Well, it just buffers whatever comes in on the modem when you are in the terminal mode. It is also a nice way to getting files from systems that don't have an XMODEM or equivalent, which was its original purpose. The 8K buffer in this version is a little on the small side. After using it a while I feel that 16K or 24K might be more comfortable, I will expand it during the next compile.

The erase, list and print commands were added as

olds to snooping. Unless the host is in your local phone area you will be using long distance. The buffer is handy for things like capturing the host's directory. Then you can get into the command mode and look at it (after the initial capture) at 9600 baud instead of 300! Even if you don't mind the cost, it's still a lot easier on the nerves. For a permanent copy, use the print command.

IT'S A WRAP

That's it for this column. I would like to give the column a new direction. As I said earlier. Time is very premium here and will be for the next year at least. I can not fit in regular columns and get any programming done. I prefer programs to writing columns and will assume that you do also. All this boils down to the fact that unless there is new software to review or other major news, I will try to bring out programs. This will keep the column active, but certainly not a monthly occurrence since most decent programs are a major undertaking.

I would welcome any programs from readers for publication. Just because you may feel that it's not earth shaking doesn't mean that it shouldn't be published. Other people will be interested in it. I have learned an awful lot from reading other people's code. Even the "simple" programs. If you don't feel comfortable sending them to me, then send them directly to the magazine. Let's get some more programs! Till then...

```

*****
*****
**
** Program:  nodedf
** Module:   nodedf.h
** Author:   N. F. Comeo
** Last edit: 2/19/84
**
** Header file for nodedf, an XMODEM compatible program
** using the Ward Christensen file transfer protocol.
** This particular implementation draws very heavily on
** progress or ideas either previously published or sent
** to the author by:
**
**      Jon Bellomo
**      David D. Clark
**      Pete Siegel
**      Sid Thompson
**
** Their contributions are gratefully acknowledged. This
** program is placed into the public domain and may be
** copied and modified without permission from the author
** providing proper acknowledgment is made.
**
*****
*****

```

#include "STDIO.H"

```

/*
** Special characters for the file transfer protocol
**
#define TIMEOUT  -1
#define SDH      1    /* ^A -- start of sector */
#define EGT      4    /* ^D -- end of transmission */
#define ACK      6    /* ^F -- successful transfer */
#define CAN      24   /* ^Z -- cancel transmission */
#define NAK      23   /* ^U -- error detected */
#define SECSIZ   128   /* bytes per sector */
#define BUFSIZ   8192  /* the capture buffer */
#define EARDMAI  10    /* number of errors till abort */
#define RETRYMAX 5     /* number of retries till abort */

```

```

/*
** Command key definitions
*/
#define BAUD      'b'
#define CAPTURE   'c'
#define DOWNLOAD  'd'
#define ERASE     'e'
#define FILESET   'f'
#define HELP      'h'
#define KEEP      'k'
#define LIST      'l'
#define PRINT     'p'
#define QUIT      'q'
#define SETTINGS  's'
#define TERMINAL  't'
#define UPLOAD    'u'
#define VIEW      'v'
#define LOCAL     0x00 /* ^@ (shift-control-2) */

```

```

/*
** Acct: defines
*/
#define PARMSK    0x7F
#define DEL       0x7F
#define BYMSK     0xFF
#define BS        0x08
#define TAB       0x09
#define LF        0x0A
#define CR        0x0D
#define CTRLZ     0x1A /* end of text-file character */

```

```

/*
** hardware defines
*/

/* acia stuff */
#define MODEN     0xE000
#define TERN      0xE004
#define DATA     1
#define CONTROL   0
#define STATUS    0
#define RDRF      0x01 /* rcv ready status bit */
#define TDRF      0x02 /* xmt ready status bit */
#define RESET     0x03 /* reset bit pattern */
#define READY     1
#define NDTREADY  0
#define B300      0x16
#define B1200     0x15
#define SPS        0000 /* should be tweaked 'per system' */

```

```

/* pia stuff */
#define PRINTER    0xE00A
#define PDATA      1
#define POIR       0
#define PSTAT      1
#define PCOMT      1
#define PM_BITS    0xFF /* write bits */
#define PSETUP     0x2E /* control setup bit pattern */
#define PRDY_BIT   0x80 /* ready bit */

```

```

/*
** Misc. defines
*/
#define DOTS       50 /* sector counting dots per line */
#define ERROR      -1
#define TABSIZ     0
#define CPM        0 /* editorial comment?? */
#define ASCII      1
#define BINARY     2
#define RA         "r"
#define RB         "rb"
#define MA         "a"
#define MB         "wb"

/*
** global variables
*/
int  txtptr, lstptr; /* index's into the capture buffer */
FILE  *fd;
char  file_name; /* file access code flag */

```



```

eofflag,      /* end of file flag */
lf_flag,      /* LF flag for CPM mode */
viewmode,     /* viewing mode */
cflag,        /* capture flag */
pflag,        /* print flag */
kbdata,       /* data from keyboard */
mdata,        /* data from modem */
asciiflag,    /* show transmitted data switch */
showtrans,    /* show received data switch */
showrecv,     /* view switch */
view,         /* read file open mode string */
or_str,       /* write file open mode string */
wv_str,       /* the capture buffer */
buf1[BUFSIZ], /* file name buffer */
filename[15], /* general buffer for user input */
answer[10];

/* =====
=====
*/
/* Program:  mode09
Module:  mode01.c
Author:  N. F. Cooso
Last edit: 2/4/84
*/
/* This is the main module. It contains the top level
command handler, terminal emulator, and system set up
functions.
*/
/* =====
=====
*/

#include "MODEM.H"

#define REVNO "1.3" /* rev number for entry banner */
#define CLRCRM "\033C" /* K19 initialization string */

static tableft, /* # how many columns are left until a tab stop */
tabsize, /* the tab column modulo */
cur_baud; /* the current baud rate */

main()
{
    /*
    * initialize program and hardware
    */
    file_mode = ASCII;
    r_str = RA;
    w_str = WA;

    viewmode = 'a';
    asciiflag = showrecv = showtrans = FALSE;
    cflag = view = FALSE;
    lctptr = lctptr = 0;
    tabsize = tableft = TABSIZE;
    initport(MODEM, B1200);
    init_pio(PRTNTR, PW_BITS, PRTUP);
    cur_baud = 1200;
    printf("IsMODEM? Rev: %s\n", CLRCRM, REVNO);
    list_settings();
    printf("\n\nHit 'h' for help\n");

    /*
    * the top most command processor
    */
    for (;;)
    {
        printf("\n>> ");
        kbdata = getchar();
        putchar('\n');
        switch(tolower(kbdata))
        {
            case BAUD:
                set_baud();
                break;

```

```

            case CAPTURE:
                set_capture();
                break;

            case SETTINGS:
                list_settings();
                break;

            case HELP:
                instruct();
                break;

            case KEEP:
                keep_test();
                break;

            case QUIT:
                hangup();
                exit();

            case DOWNLOAD:
                readfile();
                break;

            case UPLOAD:
                sendfile();
                break;

            case TERMINAL:
                terminal();
                break;

            case VIEW:
                set_view();
                break;

            case ERASE:
                erase_buf();
                break;

            case PRINT:
                print_buf();
                break;

            case LIST:
                list_buf();
                break;

            case FILESET:
                set_file();
                break;

            default:
                printf("\nUnknown command\n\n");
                break;
        }
    }

    /*
    * print out all the commands to the
    * console
    */
    instruct()
    {
        printf("Commands:\n");
        showchar(BAUD);
        printf("\t- set the mode's baud rate\n");
        showchar(CAPTURE);
        printf("\t- toggle test capture\n");
        showchar(DOWNLOAD);
        printf("\t- download a file from the host system\n");
        showchar(ERASE);
        printf("\t- erase the capture buffer\n");
        showchar(FILESET);
        printf("\t- set file mode to Ascii, Binary or CP/M\n");
        showchar(HELP);
        printf("\t- list the commands\n");
        showchar(KEEP);
        printf("\t- file the capture buffer\n");
        showchar(LIST);
    }

```

```

    printf("\t- list the capture buffer\n");
    showchar(PRINT);
    printf("\t- print, then erase the capture buffer\n");
    showchar(BUIT);
    printf("\t- return to FLEX\n");
    showchar(SETTINGS);
    printf("\t- list the current program parameters\n");
    showchar(TERM/MAL);
    printf("\t- enter the terminal mode\n");
    showchar(UPLOAD);
    printf("\t- upload a file to the host system\n");
    showchar(VIEW);
    printf("\t- toggle data viewing, initially inactive\n");
    showchar(LOCAL);
    printf("\t- enter the local mode (from the terminal mode)\n");
}

/*
 * Terminal emulator, transparent to all
 * key codes except NULL, which causes it
 * to return to the command mode.
 */
termcall()
{
    printf("Entering terminal mode\n");
    while (cترمادى)
    {
        /* handle any console input data */
        if (istat(TERM))
        {
            if (kbdata = (inchar(TERM) & PARMSK) == LOCAL)
            {
                printf("\nEnter local mode\n");
                return;
            }
            outchar(kbdata, MODEN);
        }

        /* handle any modem input data */
        if (istat(MODEN))
        {
            mdata = inchar(MODEN) & PARMSK;
            capture(mdata);
            outchar(mdata, ICRN);
        }
    }
}

/*
 * change the capture state
 */
set_capture()
{
    if (cflag == "cflag")
    {
        printf("Capture initiated\n");
    }
    else
    {
        printf("Capture terminated\n");
        printf(" %u bytes free\n", BUFSIZ - tptr);
    }
}

/*
 * keep any captured test
 */
keep_test()
{
    int i;

    if (tpttr == 0)
    {
        printf("Nothing to save\n");
        return;
    }

    prompt("Save to file: ", filename);
    if ((fd = fopen(filename, "w")) == ERROR)
        printf("Can't open %s\n", filename);
    else

```

```

        for (i = 0; i < tpttr; i++)
            if (putc(buf[i], fd) == ERROR)
                break;
        fclose(fd);
        cflag = FALSE;
        tpttr = istptr = 0;
        tableft = tabsize;
    }
}

/*
 * erase the capture buffer
 */
eras_buf()
{
    printf("Ready to clear the capture buffer.\n");
    prompt("Are you sure (y/n)", answer);
    if (tolower(answer[0]) != 'y')
        return;
    cflag = FALSE;
    tpttr = istptr = 0;
}

/*
 * list out the capture buffer
 */
list_buf()
{
    int i;

    for (i = 0; i < tpttr; i++)
    {
        putchar(buf[i]);
        if (i % 64 == 0)
            putchar(LF);
    }
    printf("\n\n");
}

/*
 * set the baud rate
 */
set_baud()
{
    unsigned rate;

    prompt("Enter baud rate: ", answer);
    rate = atol(answer);
    switch(rate)
    {
        case 300:
            initport(MODEN, B300);
            break;

        case 1200:
            initport(MODEN, B1200);
            break;

        default:
            printf("Illegal value -- baud rate not changed\n");
            return;
    }
    cur_baud = rate;
}

/*
 * list the current switch
 * settings to the console
 */
list_settings()
{
    printf("\nThe current settings are\n");
    printf("\n  Baud rate:      %u", cur_baud);
    printf("\n  capture mode:   %s",
        showbool(cflag));
    printf("\n\n  In characters are saved", tpttr);
    printf("\n  view mode:      %s",
        showbool(view));
    printf("\n  ICR, view mode: %s",
        showbool(view));
    printf("\n  file mode:      %s",
        switch(file_mode));

```

```

{
case ASCII: printf("Ascii");
break;
case BINARY: printf("Binary");
break;
case CPM: printf("CP/M");
break;
}
}

```

```

/*
 * save data to capture buffer in FLEX
 * "ascii" mode by:
 *
 * - expanding tabs to spaces
 * - swallow all control codes
 * except <cr>
 */
capture(c)
char c;
{
if (cflag && tptr < BUFSIZ)
{
if (c == '\t' && c != PARMSKI)
{
buf[tptr++] = c;
if (--tableft <= 0)
tableft = tabsize;
}
else if (c == CR)
{
buf[tptr++] = c;
tableft = tabsize;
}
else if (c == TAB)
while (--tableft >= 0)
buf[tptr++] = ' ';
tableft = tabsize;
}
else if (cflag)
{
printf("\n\nCapture buffer overflowed!");
cflag = FALSE;
}
}
}

```

```

/*
 * issue the prompt and return the
 * response, left justified
 */
prompt(p,a)
char *p, a;
{
printf("%s",p);
getchar();

/*
 * left justify the answer
 */
p = a;
whileop && (op == ' ' || op == '\t')
op++;
while (op)
a++ = op++;
a = '\0';
}

```

```

/*
 * show command characters nicely
 */
showchar(c)
char c;
{
if (c <= c && c < '\0')
printf("\t\t\t%c",c);
}

```

```

else if (c == '\0')
printf("\t\t\t");
else if (c <= c && c < '\0')
putchar(c);
else if (c == 127)
printf("\t\t\t");
else
printf("\t\t\t",c);
}
}

```

```

/*
 * show the state of a boolean
 */
showbool(b)
int b;
{
if (b)
printf("on");
else
printf("off");
}

/*
 * print out the buffer to the parallel
 * port, checking for the user input
 */
print_buf()
{
while (tptr <= tptr)
{
parout(buf[tptr++],PRINTER);
if (!instat(TERM))
continue;
kbddata = getchar();
if (tolower(kbddata) == 'q')
return;
if (tolower(kbddata) == 'e' || tolower(kbddata) == 'u')
while (!instat(TERM));
getchar();
}
}

```

```

#####
#####
** Program:  modex          **
** Module:  modex2.c       **
** Author:   M. F. Conno   **
** Last edit: 2/19/84      **
**
** This file contains the file send/receive routines for
** the Ward Christianson protocol.
**
#####
#####

```

#include "MODEM.H"

```

/*
 * read a file using the IMODEM protocol
 */
readfile()
{
int r_char, sectsize, sectcomp, sectcomp, error;
int checksum, r_checksum, errorflag, id, j;
char sector[SECSIZE], mode[20];

/* get the filename */
prompt("Enter filename ",filename);

/* set up viewing flags */
showrcv = showtrns = FALSE;
if (view)
showrcv = TRUE;
}

```



```

/* try to open the save file */
if ((fd = fopen(filename, "w")) == ERROR)
{
    printf("Can't open file for 'w'\n", filename);
    return;
}

/* now, read it in */
printf("Receiving file\n", filename);
sectnum = errors = 0;
sendchar(NAK);

do
{
    errorflag = FALSE;
    do
    {
        r_char = readchar(5);
        while (r_char != SOH && r_char != EOT && r_char != TIMEOUT);

        if (r_char == TIMEOUT)
        {
            errorflag = TRUE;
            printf("Timeout error\n");
        }

        if (r_char == SOH)
        {
            /* get sector number and complement */
            sectcurr = readchar(1) & 0xFF;
            sectcomp = readchar(1) & 0xFF;

            /* try to read the sector */
            if (sectcurr + sectcomp == 255)
            {
                if (sectcurr == (1 - sectcurr) & 0xFF)
                {
                    /* read the sector */
                    checksum = 0;
                    if (viewmode == 'a')
                        asciiflag = TRUE;
                    for (i = 0; i < SECSIZ; i++)
                    {
                        sector[i] = readchar(1);

                        checksum += sector[i];
                        checksum &= 0xFF;
                    }
                    asciiflag = FALSE;

                    /* test the checksum */
                    r_checksum = readchar(1) & 0xFF;
                    if (checksum == r_checksum)
                    {
                        errors = 0;
                        sectnum++;
                        if (!write(fd, sector, SECSIZ) == ERROR)
                        {
                            printf("Save file write error\n");
                            fclose(fd);
                            return;
                        }
                    }

                    /* now hang in there for a sender
                     * timeout
                     */
                    idle();
                    if (!showrecv)
                        printf("Received sector\n");
                    sendchar(ACK);
                }
            }
            else
            {
                printf("Checksum error\n");
                errorflag = TRUE;
            }
        }
    }
    else if (sectcurr == (sectnum & 0xFF))
    {
        printf("Received duplicate sector %d\n", sectnum);
        /* wait for idle line */

```

```

        idle();
        sendchar(ACK);
    }
    else
    {
        printf("Synchronization error\n");
        errorflag = TRUE;
    }
}

/* check for errors */
if (errorflag)
{
    errors++;
    printf("Error %d\n", errors);
    while (readchar(1) != TIMEOUT);
    sendchar(NAK);
}

while (r_char != EOT && errors < ERRORMAX);

if (r_char == EOT && errors < ERRORMAX)
{
    sendchar(ACK);
    fclose(fd);
    printf("Transfer completed\n");
}
else
    printf("Aborting\n");

/*
 * send a file using the PRODEM protocol
 */
sendfile()
{
    int sectnum, attempts, status, checksum, fd, i;
    char sector[SECSIZ], mode[20], r_char;

    /* get the filename and file mode */
    prompt("Enter filename ", filename);

    /* do some housekeeping */
    showrecv = showtrans = FALSE;
    if (!view)
        showtrans = TRUE;

    if ((fd = fopen(filename, "r")) == ERROR)
    {
        printf("Can't open file\n", filename);
        return;
    }
    eofflag = iflag = FALSE;

    printf("Sending file\n", filename);

    sectnum = 1;

    /* wait for initial NAK */
    attempts = 0;
    while (readchar(5) != NAK && attempts < RETRYMAX)
        attempts++;
    if (attempts >= RETRYMAX)
    {
        printf("Timed out waiting for initial NAK\n");
        fclose(fd);
        return;
    }

    /* send the file */
    attempts = 0;
    while (attempts < RETRYMAX)

```

```

{
    status = fread(&id,sector,SECSIZ);

    /* exit on EOF or ERROR, we can't tell which */
    if (status == EOF)
        break;

    /* try sending the sector a few times */
    attempts = 0;
    do
    {
        if (!showtrans)
            printdot(sectnum);
        sendchar(SOH);
        sendchar(sectnum & 0xFF);
        sendchar(sectnum > 0xFF);
        checksum = 0;
        if (viewmode == 'a')
            asciiflag = TRUE;

        /* send the sector */
        for (i = 0; i < SECSIZ; i++)
        {
            sendchar(sector[i]);
            checksum += sector[i];
            checksum &= 0xFF;
        }

        asciiflag = FALSE;
        sendchar(checksum);
        attempts++;

        /* get receiver's response to the block */
        purgeline(MODEM);
        r_char = readchar(5) & PARMSK;
        if (r_char == MAX)
            printf("MAX\n");
        else if (r_char == TIMEOUT)
            printf("TO\n");
        else if (r_char == ACK)
            printf("%02x",r_char);
    }
    while(r_char != ACK && attempts < RETRYMAX);
    if (attempts < RETRYMAX)
        sectnum++;
}

/* check for errors now */
if (attempts == RETRYMAX)
    printf("\nNo acknowledgement of sector -- Aborting!\n");
else
{
    attempts = 0;
    do
    {
        sendchar(ETB);
        purgeline(MODEM);
        attempts++;
    }
    while (readchar(5) != ACK && attempts < RETRYMAX);
    if (attempts == RETRYMAX)
        printf("\nNo acknowledgement of End Of File -- Aborting!\n");
    else
        printf("Transfer complete!\n");
}
fclose(fd);
}

/*
 * timed read of the modem port
 */
readchar(seconds)
    unsigned seconds;
{
    char data;

    seconds = seconds & 0xFF;

    /* wait for data */
    while(!instat(MODEM) && seconds)
        seconds--;
}

```

```

/* test for timeout */
if (seconds == 0)
    return TIMEOUT;

/* get data and show it if necessary */
data = (uchar)MODEM;
if (showrecv)
    showdata(data);
return data;
}

/*
 * send a char
 */
sendchar(data)
    char data;
{
    /* wait until output is ready */
    while (!outstat(MODEM));

    /* send the data */
    outchar(data,MODEM);
    if (!showtrans)
        showdata(data);
}

/*
 * show a character
 */
showdata(data)
    char data;
{
    if (asciiflag)
        if ((data >= ' ' && data <= PARMSK) ||
            data == LF || data == CR || data == TAB)
            putchar(data);
        else
            printf("%02x",data);
    else
        printf("%02x",data);
}

/*
 * print a dot on the screen
 */
printdot(nval)
    int nval;
{
    if ((nval - 1) % DOTS == 0)
        printf("\n(%d)",nval);
    else
        putchar('.');
}

/*
 * set up the view state
 * MODEM transfers
 */
set_view()
{
    char answer(20);

    if (view == "view")
    {
        prompt("View as Ascii or Hex: ",answer);
        printf("Viewing will be in ");
        if (tolower(answer[0]) == 'h')
        {
            viewmode = 'h';
            printf("Hex\n");
        }
        else

```

```

        {
            viewmode = 'a';
            printf("Ascii\n");
        }
    }
    else
        printf("Viewing Disabled\n");
}

/*
 * wait for an idle line
 */
idlein()
{
    int n;
    {
        while (readchar() != TIMEOUT);
    }
}

/*
 * set up the file characteristics
 * for Xmodem transfers
 */
set_file()
{
    prompt("Enter file mode (Ascii/Binary/Cpo)", answer);
    switch(tolower(answer[0]))
    {
        case 'b': file_mode = BINARY;
                  r_str = RD;
                  w_str = WD;
                  break;

        case 'c': file_mode = CPO;
                  r_str = RA;
                  w_str = WA;
                  break;

        default:  file_mode = ASCII;
                  r_str = RA;
                  w_str = WA;
                  break;
    }
}

/*****
*****
**
** Program:   modem9
** Module:    modem3.c
** Author:    M. F. Cosmo
** Last edit: 2/19/84
**
** This file contains the modem hardware drivers, and the
** real disk I/O routines.
**
*****/

#include "MODEM.H"

/*****
*****
**
** acia drivers
**
*****/

/*
 * initialize the ACIA hardware
 */
initport(port, setval)
char port[];
int setval;
{
    port[CONTROL] = RESET;
    port[CONTROL] = setval;
    purgeline(port);
}

```

```

/*
 * test for an input character
 */
instat(port)
char port[];
{
    return (port[STATUS] & RDRF);
}

/*
 * get an input character
 */
inchar(port)
char port[];
{
    while (!instat(port));
    return port[DATA];
}

/*
 * test for output ready
 */
outstat(port)
char port[];
{
    return (port[STATUS] & TSRE);
}

/*
 * send a character
 */
outchar(c, port)
char c, port[];
{
    while(!outstat(port));
    port[DATA] = c;
    return c;
}

/*
 * purge the input line of data
 */
purgeline(port)
char port[];
{
    while(instat(port))
        (inchar(port));
}

/*****
*****
**
** pia drivers
**
*****/

/*
 * initialize the pia
 */
init_pia(port, bits, setval)
char port[], bits, setval;
{
    port[PCONT] = 0;
    port[PDIR] = bits;
    port[PCONT] = setval;
    port[PDATA] = '\0';
}

/*
 * send a char to the port,
 * expanding <cr> into <cr>X(14)
 */
parout(c, port)
char c, port[];
{
    if (c == CR)

```

```

        poutd(LF, port);
        poutd(c, port);
        return c;
    }

/*
 * actually send the character
 */
poutd(c, port)
    char c, port[];
{
    char junk;

    while (parstat(port) == NOTREADY);
    junk = port[PDATA];
    port[PDATA] = c;
}

/*
 * check the port for ready
 */
parstat(port)
    char port[];
{
    return (port[PSTAT] & PRDY_BIT);
}

/* =====
 *
 * More are the disk routines that do a
 * reasonable job of simulating low
 * level disk I/O
 *
 * =====
 */

/*
 * Try to read a buffer from the disk;
 * if less, then NULL pad the rest of
 * the sector
 *
 * Special dispensations for CPM (ugh)
 * 1) expand newline into a <cr><lf>
 * 2) pad partial sector with ^Z
 * 3) tag a sector of ^Z's on the end
 * just in case
 */
fread(f, b, n)
    FILE *f;
    char ob;
    int s;
{
    int bytent;
    int c;          /* must be an int to preserve EOF */

/*
 * if EOF and CPM then kick out one
 * more sector of eof's
 */
if (!eofflag)
{
    if (file_mode == CPM && eofflag < 2)
    {
        bytent = 0;
        goto pad_sector;
    }
    return EOF;
}

/*
 * special case for the CPM mode when '\n'
 * was the last character of the last sector,
 * so we have to pick up in this sector
 */
bytent = 0;

```

```

if (file_mode == CPM && lf_flag)
{
    ob++ = LF;
    bytent++;
    lf_flag = FALSE;
}

/*
 * read in a sector, one byte
 * at a time
 */
for (; bytent < s; bytent++)
    if (c =getc(f) == EOF)
    {
        eofflag = TRUE;
        break;
    }
else
    switch (file_mode)
    {
        case BINARY:
            ob++ = (char)c;
            break;

        case ASCII:
            if (c == '\n')
                ob++ = CR;
            else
                ob++ = (char)c;
            break;

        case CPM:
            if (c == '\n')
            {
                ob++ = CR;
                if (bytent < (s - 1)) /* room for <cr><lf> ? */
                {
                    ob++ = LF; /* yes */
                    bytent++;
                }
            }
            else if (lf_flag == TRUE) /* save for next time */
            {
                lf_flag = FALSE;
            }
            else
                ob++ = (char)c;
            break;
    }

/* pad if not full */
pad_sector:
switch (file_mode)
{
    case CPM:
        while (bytent < s)
            ob++ = CTRLZ;
        break;

    default:
        while (bytent < s)
            ob++ = NULL;
        break;
}
return bytent;
}

/*
 * write out a sector
 */
fwrite(f, b, n)
    FILE *f;
    char ob;
    int s;
{
    int bytent;

    for (bytent = 0; bytent < s; bytent++)
        if (putc(ob++, f) == ERROR)
            return ERROR;
    return bytent;
}

```



# WHERE IS ? ? THE 68000 ?

Where is the 68000? That is a question we have heard often from the '68' Micro Journal Readers. Most of our readers expected to see the 68000 appear in SS-50 Bus machines before any place else for several reasons: the close ties to the Motorola CPU Family; the fact that all that was needed, initially, was a simple Operating System and a CPU Card (all of the I/O links, etc., were already fairly compatible), along with a simple adaptation of Motorola's Assembler, to allow serious development work to begin; the fact that, while the 68000 architecture is different from the 6800/6809, it is closer in both Hardware and Software to them than is any of the '80 Series CPUs; etc. The Software Writers, especially, who have doggedly supported the SS-50 Bus, and who have longed for a viable Marketplace for their Products, felt a rise of joy with the 68000's announcement; maybe, AT LAST, a Motorola CPU that the whole WORLD acknowledged as "THE" CPU of the future, would provide the impetus to get the SS-50 Bus Systems out into the Real World in numbers that would make supporting them worth all of their effort. While most of these Writers (AND the '68' Micro Readers) are STILL waiting for SOMETHING from the SS-50 Bus Manufacturers, those that are REALLY interested in the 68000 have begun to investigate other avenues.

Yes, there are a few SS-50 Bus 68K Boards that will appear in the near future. The LSI 68008 Board is about to be released (w/ CP/M 68K); Glmix expects to have a 68000 CPU Card for their Glmix III System late this Summer (w/ Microware's 68K Ver. of OS-9); and Hazelwood should have their 68008 Board available this year (probably w/ CP/M 68K); but I am afraid that they will be too late to have the Market Impact that would induce "droves" of Software Writers to write the multitude of Programs that are REQUIRED to make these Systems a viable force in the Market Place of Today (AND generate enough Software Sales that a Writer can make a LIVING by supporting that System).

The overall feeling among the SS-50 Bus Software Producers is one of FRUSTRATION; many are barely meeting the FIXED costs of producing Software, let alone make a small profit to help support their "Hobby" (and some, such as Howard Harkness of Words Worth and Gerry Williams of JPC, have simply given it up as not worth the effort of taking orders, shipping, answering questions, keeping up with who has what version, wrestling with the financial problems such as telephone bills, printing costs, credit card discounts, etc.). Is it worth even a hobbyist's time to spend three years developing a program that probably won't sell a thousand copies over the life of the System it is to run on. If a piece of Software for an Apple II, or a Commodore, or Atari, or IBM PC, etc., doesn't sell a thousand a month, it is dropped from the Market Place as a total loss. WE know that the quality of the SS-50 Bus Machines are magnitudes better than the "competition", and that the Operating Systems are "un-touchable", but WHERE are they in the Market Place of Today?

Everyone has been waiting for a 68000 Operating System to emerge as a "front runner". While the field is still open, Digital Research is making headway with their CP/M 68K, and numerous "Unix Look-alikes" are appearing, as is Unix V and VII with full AT&T Support. CP/M 68K carries the CP/M "logo", which makes it appear attractive, but it is a Single-User Operating System with no more REAL Software Support than any

other DOS (obviously, none of the '80 Software will run with it until it is converted to 68000 Code). Unix, which was developed BY Programmers FOR Programmers, is becoming the "IN" Op Sys because the people who WRITE Programs like it, but it will require a new Shell (or maybe a "pre-Shell") which is magnitudes more User Friendly before it will become a real force in the "Numbers Game", or find common User Acceptance. And BOTH are written in C.

While C is a MAJOR improvement over most Compiled Languages, and should find a home in many Application Software situations, it is still not the answer in a "Real-Time" Interactive User Environment such as a Word Processor or an Operating System on a Computer "For the Masses" until we get away from BOTH the "Ron Cain Small C" AND the Mainframe types of Compilers, and develop a C Compiler that is STREAMLINED for the specific CPU it is to be used on. C is not as fast as Assembly Language, and eats up a lot more Memory (which, while not as expensive as it used to be, is still a major portion of the cost of a Computer System). A simple C Program to count the number of characters, words, or lines in a file produces a Command File size of 7228 bytes (Microware's "count" utility), while the whole OS-9 Assembler has a File Size of only 5844 bytes. As for speed, the hesitant start of the original Apple Lisa should provide a fairly good indication of its value in an Operating System. While Ron Cain's "Small C Compiler" was probably the MAJOR force in making C a viable Language in the Microcomputer Arena, it was written for an 8080, and uses an Intermediate Register Set of ONE 8-Bit Register and ONE 16-Bit Register; obviously, both the 6809 and 68000 have a little better capability than THAT (Re. "C Compiler Techniques" by Bud Pass in the Nov. '83 "System-68" - I think it was the final issue?). The "Mainframe-type" Compilers could care less about the amount of code generated; memory is the LEAST of their problems. For another indication of the power of speed and efficiency in the Micro Market-place, look at the BDS C Compiler. It was "streamlined" for a Z80 and CP/M, and while it is not even CLOSE to "K&R" (try converting a BDS C Program to run on a Unix Compiler), it Compiles fast and produces efficient code on a Z80 CP/M System, and has become a real FORCE in the CP/M C Compilers over the "standard" Compilers; again, speed and efficiency wins out.

Where ARE the 68000 "Machines"? Basically, they are just about everywhere BUT on the SS-50 Bus!

There are some interesting Non-SS-50 Bus 68000 Systems in the Market, along with a multitude of "Standard" 68000-based Computer Systems. For the Hobbyist, Sinclair Research Limited (50 Stamford St., Boston, MA 02114 617-742-4826; drop them a note, or give them a call -- tell them WE sent you) has a 68008 Computer System (the Sinclair QL) for \$499, less a Display Unit. Their original ZX80 set the Computer Industry on its ear a few years ago; look at the specs on the "QL". For \$499 (as of now -- a US Version should be available this Summer), you get:

- a 7.5 MHz 68008;
- 128K Memory, expandable to 640K with the .5MB RAM pack (32K dedicated to the Screen bit map);
- TWO built-in 100KB "Microdrive" Tape Units with 3.5 second average access time and 15 bytes/sec transfer rates (expandable to 8 Drives);
- 32K ROM which contains Sinclair SuperBASIC featuring procedure structuring, syntax extensibility, interpretation speed independent of program size, and Operating System Support; and the Sinclair QDOS Operating System which features such things as Single-User, MULTI-Tasking, Display handling for multiple Screen Windows, and Device-Independent I/O
- a second processor (Intel 8049) which controls the Keyboard, Sound, an RS-232-C Port, and real-time Clock;
- BOTH RGB Monitor and a TV Output -- the RGB provides 512 x 256 pixels in 8 colors with an 85 x

25 character display format, while the TV output provides 40 to 60 columns, depending on software;

- full 65 key QWERTY Keyboard with 5 function keys and 4 cursor-control keys;
- two joystick ports
- and **FOUR SOFTWARE Packages**, including
  - QL QUILL - a word processor,
  - QL ABACUS - an electronic spreadsheet,
  - QL ARCHIVE - a database program, and the
  - QL EASEL - graphics package.

And that is only a SUMMARY of the Sinclair QL's features. We will provide more information as it becomes available (and any of you Readers that have seen one of these machines, let us know about it!).

For the "Experimenter Types", another interesting 68000-based Computer System is the "dimension", from Micro Craft Corp., 4747 Irving Blvd., Dallas, TX 75247 (214) 630-2562. The dimension is an 8 MHz 68000-based Computer System (physically similar to an IBM PC, including a detachable keyboard) that comes with the CP/M 68K Operating System, 128Kb memory, two 5 1/4" double-sided Disk Drives, an RS-232C port, a parallel port, a graphics controller, a real-time clock, and a 6-slot expansion bus for \$3995. What's different about it? First, the expansion slots have ALL of the 68000's lines, internal memory access and control lines, disk drive lines, etc., so that various emulator boards can be plugged in that have control of the complete Computer System. Boards are now available to allow the dimension to run IBM PC Software, Apple II Software, Radio Shack Model III/IV, etc. Each Board comes with the appropriate CPU and specific emulation Software, and is priced at \$495. The Overall System was designed specifically to provide these "emulation" capabilities, thus the expansion slot provisions. The dimension provides 64Kb of graphics memory which yields a bit-mapped resolution of 160x500 in 16 colors to 640x500 pixels on/off, the capability to define as many character sets as you have memory for and use any of them "on the fly", define display scroll areas through low-memory pointers, etc. The Keyboard "Key Codes" are accessible through a pointer in low-memory, allowing User Control of these, also. Special Utilities are provided which allow things like changing the Display Format from a 20x20 to a 100x25 character display, or select from numerous Disk Formats (or define your own, if you want), etc. Another unique feature of the System is that there are NO Seals to break to get INSIDE the Computer; in fact, Micro Craft ENCOURAGES it. They also give a ONE YEAR Warranty.

Finally, for "every John Doe in the World", there is the Apple Macintosh. If you don't know about the \$2500 "Mac" by now, you have had your head in the sand somewhere, so I won't go into all of the details of it here. I will say that it appears that it WILL become a real force in the Microcomputer World. For one thing, Microsoft expects HALF of their INCOME the next year to be from Macintosh Software. When you consider their income from the different BASIC's, MS DOS, their Compilers, etc., that is big expectations from a little Computer. Apple indicates that they have commitments from over 100 Software Developers and Suppliers for support of the Mac. They are also going ALL OUT to support valid Software Developers (I believe you could almost talk them out of the Source Code for the ROM if you tried). If you have not actually SEEN one yet, then you had better take a hard look; those "Windows" are FAST, SMOOTH, and Operationally EFFICIENT, and caused ME, an long time "Keyboard Pounder", to set up and take notice. How long did it take you to become comfortable with FLEX, let alone OS-9? I, personally, have had MY plate full of crow; I am now ready to take a realistic, hard look at the Macintosh/Lisa combination. I think the combination may offer some real possibilities.

Much has been said about the Mac's lack of expansion ports; that the interconnection between the Mac and Lisa is not a true Multi-User Interface.

True! But, tell me WHY a System has to share a CPU to be USABLE! What has to be shared is the DATA, not the CPU; that is why Networking is becoming so popular. I believe that the Mac/Lisa combination will provide a means for the "World" to break away from the "Old Ways" and allow shrewd Software Designers to develop new methods and techniques which may provide BETTER solutions to Application Problems, such as multiple-access Data Base use, for example. After all, the slate is CLEAN, as far as these machines are concerned. No one has a Software Base that can be simply "moved over" to the Mac or Lisa; they are not '80 type machines that are still running 8080 Code on an 8086 or 80286, using the same old non-Window "System Calls". Sure, I expect the early Mac and Lisa Software to be "re-written" interpreted or Compiled ports from what is now the "IN" Programs; but Apple will sell enough of these Computers to make the effort required to design "New" Products worth the effort, and the speed and efficiency of a well-designed Assembly Language Program will make the effort worth the writers time. If Mac does nothing more than help break some of the "Computer Science World" out of the 8080 rut, it will have accomplished more than any other product has ever accomplished.

We at Computer Publishing Inc., along with our Readership, consider the 68000 to be part of OUR stable. We, like others, thought that it might be the "salvation" of the SS-50 Bus, and cheered its appearance. While we are in NO way abandoning the SS-50 Bus or the 6800 and 6809, we ARE looking for 68000 information, no matter WHAT Bus or Operating System it is on. As any regular reader of the '68' Micro Journal is well aware, WE here in the Office do not WRITE the Magazine; we simply pass on what we receive from the Readership. I can assure you that we have published just about EVERYTHING we have received that had ANY reference to the 68000.

It may appear that we are heading in another direction with this statement, but that is not our intent. Computer Publishing Inc. has supported the "Motorola Microprocessors", which, until the Color Computer came along, meant that we primarily supported the SS-50 Bus Systems. The 6809 is an excellent Microprocessor that, in many situations, is the BETTER processor for an application, and it will be interesting to see where it finally "settles in" in the Microcomputer Industry.

But the 68000 is appearing in the Applications Market on a variety of Bus Structures and with a variety of Operating Systems. We hope that YOU will use the '68' Micro Journal as a "forum" for discussing the pros and cons of ANY 68000-based Computer System, Operating System, Program, etc., with the overall objective of spreading the information around. Obviously, no ONE User can be in a position to compare ALL of the Systems that are available, so pass on what YOU have so that others can learn from it. This is far and away the most efficient way to help the 68000 progress into the types of Systems that we all hope will develop over the years.

We have a unique opportunity to help "guide" the 68000's future. YOUR comments will provide guidelines for the Hardware and Software Producers; some "brave souls" have taken a first shot, now they NEED to KNOW what you like and dislike about these Systems. What are the GOOD and BAD points of an Operating System. What do you LIKE about THIS particular Product, and what would you like to see changed. Most Readers are familiar with FLEX and OS-9; how does Op Systems like CP/M 68K, PICK, Unix, Xenix, Oasis, etc., etc., etc., compare to them (since they provide a common reference). What are the good and bad points, expansion capabilities, prices, etc., of Systems such as SAGE, CODEX, Mac and Lisa, Multibus, Versabus, etc. YOUR comments provide information for both the Manufacturers AND the Users. If you have seen a new 68000 Product, let us know about it.

Finally, the 68000 provides a new and flexible

architecture. How can it best be used? What is similar, and what is different, between it and the 6809 as far as Programming Techniques, Register Manipulations, Parameter Passing, Interrupt Processing, Stack Manipulations, etc., is concerned? What are the most efficient algorithms for the 68000, whether it be Task Swapping, Graphics Display manipulations, List indexing, or what ever?

Where is the 68000? Its out there; in everything from the Sinclair QL, through the IBM 9000, and beyond. It is finding heavy use in the CAD/CAM industry. Just because it is not on the SS-50 Bus doesn't mean the Readers are not interested in it. Sure, owners of present SS-50 Bus Systems are anxiously awaiting those 68000 CPU Cards, because a CPU Board is a lot less expensive than a whole new Computer System. But, that does not mean that that is the ONLY System they are interested in. Let us hear from you if you have ANY 68000-based System!

— RLM —

## A REVIEW PROGRAMMING IN C

by Leroy Foster

I am a touch typist. Every once in a while I carelessly let my hand "shift left one bit" from the home keys. The output from the next few keystrokes is gibberish with stars, exclamation points, and other assorted garbage. Unfortunately, this was also my first impression of the C language. I purchased Kernighan and Ritchie's "The C Programming Language" and found it too esoteric. I remained C illiterate, vowing that someday -

That day is here. Stephen G. Kochan has lifted me into the ranks of C programmers with his book "Programming in C" (Hayden, 1983). Chapters 1 and 2 are introductory chapters. C starts in chapter 3. I was elated to find that I could read his sample C programs and that I could actually write a C program when he asked me to at the end of the chapter. I was on my way, devouring each chapter, learning step-by-step the ins and outs of this elegant and powerful language. Each chapter has clear examples and explanations that move from the simple to the complex. Exercises are provided at the end of each chapter to "burn in" the material. Mr. Kochan has provided 17 chapters and 6 appendices to guide the reader through C. Chapter titles include Program Looping, Making Decisions, Arrays, Functions, Structures, Character Strings, Pointers, Operations on Bits, and Input and Output. In addition, Mr. Kochan has tactfully included pertinent references and ties to the UNIX(tm) operating system.

I feel that "Programming in C" can be used by two groups of readers. If a reader has some programming experience, that reader can move at his/her own speed in a self-study mode. New programmers, the green, un-bent kind, would probably be best served in a structured

classroom setting where the book should serve as an excellent text.

Mr. Kochan recommends that the reader have access to hardware that can run both the sample programs and the exercise answers. I concur. It is the only effective way. Otherwise it is like learning to swim without water. But don't let that stop you from getting a copy of "Programming in C" if you are interested in C. Stephen Kochan has provided the incentive in his book to get you out of the armchair and up and running C.

# THE GIMIX III SYSTEM

by Robert L. Nay

## Introduction

The Computer Systems produced by GIMIX, Inc. have withstood the 'Test of Time' in the hands of users from the Personal Computer level up through the Professionals such as NASA, DuPont, Atomic Energy Commission, Ford, GM, Georgia Tech, etc., etc., etc. Standard features such as a heavy-duty Power Supply which utilizes a ferro resonant transformer, all Gold Plated Connectors, and high quality PC Boards, in combination with "State of the Art" Engineering and the constant development of new features to maximize the utilization of available Operating Systems have kept them in the fore-front of the SS-50 Bus Systems. The "motto" at GIMIX, Inc. appears to be "Uncompromising QUALITY".

The latest offering from GIMIX consists of their "GMX III" Series. Basically, the "GMX III" System consists of the GMX 6809 CPU III CPU Card, the #68 DMA Disk Controller, the #11 Intelligent Serial I/O Processor Board, and, probably the MOST important part, the GIMIX 64KB and/or 256KB Static RAM Cards, along with a choice of FLEX or, more likely, OS-9 GMXIII (a 'fine tuned' OS-9 Level II) or UniFLEX GMXIII (also specially 'tuned' for this powerful System). In addition, GIMIX offers either a 19MB or a 47MB Winchester subsystem, various combinations of 5" and 8" Disk Systems, and an Intelligent Parallel I/O Processor Board. Upgrading a present System consists of installing the appropriate Boards in the standard "Ghost Classy-Chassis", or complete Systems can be purchased beginning at less than \$6000 plus a Terminal.

The System we have been using here in the Office was upgraded from a GMX II System a few months ago, and consists of the

- GMX CPU III;
- the DMA Disk Controller driving a pair of 8" Double Sided Disk Drives, a 40 Track Double Sided and an 80 Track Double Sided 5 1/4" Drives;
- the 19MB Hard Disk System (the two 5" Disk Drives are Thin-Lines, so they set side-by-side with the Winchester in the standard Dual 5 1/4" Mounting Hole in the Chassis);
- 4 GIMIX 64KB Static RAM Cards and one GIMIX 256KB Static RAM Card;
- two Intelligent Serial I/O Processor Cards;
- an old "standard" dual-port Parallel Card;
- a Missing-Cycle-Detector Module (not used with the GMX III OS-9); and
- a Baud Rate Card.

We have the OS-9 GMX III Operating System with the new Support ROM on the CPU Card, and have two Terminals on-line at the present time (normally running Stylo simultaneously).

First, some general comments on the System.

Overall, the System performance and operation is excellent. After spending some twenty years in servicing complex electronic equipment, I am amazed at the lack of problems encountered with this System. The "secret" to the success of the GMX III System is the Static RAM Boards that they STRONGLY suggest you use; In fact, we have not found ANY Static RAM Boards that would work with this System EXCEPT the GIMIX Static RAM Cards, and we have tried many of them. We have heard a few rumblings from the field in reference to "System glitches", "Operating System burps", etc., with this System; Invariably, these Users were NOT using the GIMIX RAM (some other Boards appeared to work, but a glitch here, or one there, disappeared when those Boards were removed). If you are going to buy a race horse, don't hitch it to a wagon and expect to be happy with the results!

In the area of Software, we have had no unusual problems with Version 1.2 of the GMX III OS-9 (Level II) Operating System, and few problems with application Software. We are working with Great Plains in getting rid of a few "knit picking" problems with Stylo (which is intimately tied into the Hardware and Operating System, both of which contain many new features, and, therefore, present new problems to be solved); I have not had time to "live" with Screditor III on the System yet, but it is functional; Just about anything else we have tried appears to run OK.

#### The GMX 6809 CPU III

To quote the GIMIX Manual, "The GMX 6809 CPU III is an advanced microprocessor board, specifically designed for use with multi-user/multi-tasking operating systems". While the 6809 is an exceptionally powerful 8-bit microprocessor, it lacks some of the important features needed in a multi-user/multi-tasking environment. GIMIX has added numerous Hardware features in this CPU Card to improve system security, overall system performance, etc. The System can access 1MB of Memory in 2K segments, and each segment can be assigned certain "attributes" (such as preventing one User access to another Users space, Write Protecting specific segments, etc.). These attributes are controlled by Hardware external to the CPU itself, so a "run-away" Program, for example, will not "crash" ANOTHER Users Program.

The GMX 6809 CPU III provides some of the 68000's built-in capabilities of separating the systems operation into a Supervisor State and a User State. Any "trap" (IRQ, FIRO, the SWIs, or one of the three "built-in" traps - UAM (UnAllocated Memory), WPT (Write Protect), or WDC (WatchDog Counter)) immediately forces the System into the Supervisor Mode, which allows the System to take appropriate action. This provides several benefits; one Users' "crash" will normally have no effect on other Users, all of the trap vectors are in the Supervisor Area (normally Task No. 0) so that each user has a FULL 64K of Memory with no "system" overhead, etc.

The System utilizes DMA Transfers extensively. All I/O to and from the "Smart I/O Boards" and the Disk Controller is DMA'd, as well as Memory to Memory transfers, which are handled with the On-Board Direct Memory Access Controller (DMAC). The DMAC is capable of moving data between memory locations at 1 byte every 2 machine cycles (1 byte/microsecond at 2MHz), which greatly improves System operation (and is one of the reasons the GIMIX Static RAM Boards are REQUIRED). During DMAC transfers, the CPU is put in the "Halt" mode (i.e., the DMAC uses the "halt/burst" transfer system).

Obviously, with both On-Board and External DMA Systems operational in the GMX III System, some kind of a DMA Contention scheme is required. An On-Board Hardware DMA Arbitration system extends priority to an External "cycle/steal" DMA System such as the #68 DMA Disk Controller, but allows the DMAC system priority over any other "halt/burst" DMA System. "Smart I/O Board" transfers are accomplished by the

DMAC System.

Specific features of the GMX 6809 CPU III Board include:

2 Mhz 6809E CPU.

Expanded memory management (DAT) which supports up to 1 M Byte of memory in 2K segments.

Allows a full 64K of RAM in a Users Address Space with NO System overhead.

Memory segment attributes that provide;

Trapping of out-of-range memory references,  
Write Protection of individual segments,  
Protection against certain illegal instructions,  
and Hardware Single-Step capability.

High-Speed Memory-to-Memory and Memory-to-I/O DMA transfers at 1 byte per microsecond.

Arbitrates contention between the on-board DMAC and external DMA devices.

Separate Supervisor and User "states" to prevent unauthorized access to system information and devices.

Automatic switching to the Supervisor State in response to hardware and software interrupts.

Full function Time-Of-Day clock with year, automatic leap year/daylight savings time correction, and on-board battery.

6840 Programmable Timer with a separate 500 KHz precision (.0025%) time base oscillator.

Accepts one 2K, 4K, or 8K EPROM for on-board firmware, up to 4K mapped into the address space at one time.

Software or Hardware selection of the upper or lower half of an 8K EPROM for software switching applications.

2K CMOS scratchpad RAM with battery backup.

Numerous Jumper Selection Options are provided on the GMX 6809 CPU III Board which provide flexibility in the use of the Board for different Operating Systems, as well as providing the User with some capabilities in selecting other functions.

#### DMA Disk Controller #68

The DMA Controller leaves the processor free to perform other tasks once a transfer has been initiated, allowing the overall System to realize the full potential of today's modern Operating Systems. The DMA Disk Controller provides the features:

HIGH SPEED using bi-polar logic DMA circuitry for guaranteed operation at 2Mhz. DMA transfers take place at full bus speed using the 6809 "cycle steal" transfer mode. Once the required parameters are passed to the Controller and DMA transfer is initiated, the processor is free for other tasks. Interrupts can be generated to indicate the completion of the transfer.

SINGLE and DOUBLE DENSITY data storage on any combination of 5 1/4" and 8" floppy drives; single and double sided, single and double track density (i.e., 40 or 80 track), up to 4 Drives total.

LOW ERROR RATES are insured by a phase lock data recovery circuit (data separator) and adjustable write precompensation circuitry for drives that require precomp. Separate precomp adjustments are provided for 5 1/4" and 8" drives.

ADDRESSABLE to any 8 byte boundary in the address space (1M byte when extended address decoding is used). The Board occupies only .8 bytes of address space.

EXTENDED ADDRESSING control using the SS-50C extended address lines. Control of the extended address lines allows the board to perform DMA transfers to and from any address in the 1M Byte address space.

FULLY BUFFERED with separate 5 1/4" and 8" output buffers and Schmidt Trigger Input buffers for the disk drive signals.

The DMA Disk Controller Board also has several Option Jumpers which allow the User to configure the unit to fit his specific needs. While GIMIX Floppy Disk System operation is not as fast as some of the other



Systems, I can not remember EVER having a "crashed" Disk in over two years of using this Controller. The "slow-down" is NOT caused by the Controller, but by the Disk I/O Software and the numerous checks made on each transfer. The Floppy Disk System JUST PLAIN WORKS!

#### The GIMIX Intelligent Serial I/O Processor Board

The GIMIX Intelligent Serial I/O Processor Board is a 30 Pin Card which provides three (3) RS-232C Serial Ports. It includes a 68B09 CPU, a Z8038 FIFO, and three 6551A's, and can utilize up to 16K Memory, of which 8K can be RAM. The Intelligent Serial I/O Card can relieve much of the CPU's work load, again providing a more efficient overall System, by buffering data transfers and performing a certain amount of data preprocessing.

The GIMIX Intelligent I/O Boards are a new product to the normal SS-50 Bus Software Producers, so the overall effect of these units on system through-put can not be evaluated at this time. A combination of a "smart" line-oriented Terminal and the "smart" Serial Card should be a big help in programs like Word Processors, Electronic Spreadsheets, etc., once the Software has been converted from "character oriented" systems to "line oriented" systems. These new boards have been the source of most of the Software "bugs" noted originally.

#### GIMIX 19MB Hard Disk System

The GIMIX 19MB Hard Disk System consists of a 5.25" Winchester Hard Disk (which is the same size as a standard width 5 1/4" Disk Drive) and two 50 Pin Slot PC Boards; a Xebec SI410 Winchester Disk Controller Board is mounted on one of the SS-50 Bus Boards. If you have not had the pleasure of using a reliable Hard Disk, you are in for a shock. Disk accesses and transfers are FAST; Stylo's approx. 90 Sectors loads and initializes in about seven (7) SECONDS. In about two years of operation with this unit, we have only had one small problem (this was one of the original units). It, like the Floppy Disk System, JUST PLAIN WORKS!

The Hard Disk formats with 58,752 Sectors when formatted with the GMX III OS-9 System, which still utilizes the space as single sector "clusters". The biggest problem I had using the Hard Disk with FLEX was in trying to find a meaningful Filename; OS-9, with its Directory System, has obviously eliminated that problem.

#### Summary

The overall GIMIX III Computer System is a "state of the Art" system which combines Hardware capabilities and the power of the 6809 CPU to provide better utilization of the powerful Operating Systems available for this CPU. As usual, Hardware Design has moved ahead of Software Design; once the Software producers have a chance to "fine tune" their Software to the specific capabilities of the GMX III System, still further improvements will be realized.

Finally, the GMX III System is not a "dead end" System. GIMIX has not officially announced their 68000 Board yet, but it will be designed as a direct replacement for the 6809 CPU Card and use all of the existing boards in the III System (In fact, the "Smart" Boards, with the 68B09, actually provide faster data transfer than the 68000). The CPU will be a full 10 or 12 MHz 68000 (unlike the 68008 Boards that will be available for the SS-50 Bus shortly), and will use Microware's OS-9-68K (??? OS-9 for the 68000). Hopefully, GIMIX will provide some type of switching so that BOTH Boards can be installed in the System, allowing the use of present Software on the 6809 while Software is being developed for the 68000.

## REMOTE PULSE WIDTH

#### Remote Pulse Width Determination

An MC146805E2 CMOS microprocessor with EPROM monitor was interfaced to an Addressable Asynchronous Receiver/Transmitter. The microprocessor is configured by firmware to serve as a pulse width determination slave. The microprocessor divides its system clock by five and starts counting these pulses on the rising edge of the pulse to be measured. Two Schmitt triggers are used to trigger on the falling edge of the pulse to be measured. The output from the second Schmitt trigger is normally high and a low going pulse is applied to the IRQ pin when the falling edge of the pulse to be measured is detected. A IRQ interrupt is then generated and the timer data register is read and its contents output to the AART via port A of the MC146805E2. Firmware prevents an update of port A during an AART transmit. The AART in turn relays the hexadecimal value of the pulse width to the serial board on port 4. This transmission occurs at 4800 baud with an eight bit word, even parity, and one stop bit format. The EPROM monitor program SLAVE loops until it receives divide by information from MASTER which is running on the mainframe. The operator can divide by 1,2,4,8,16,32,64, or 128 by sending out the appropriate bit pattern by means of the AART command word to the appropriate port B inputs of the microprocessor. These bits are then shifted and AND'ed with a byte that will set up the timer control register to operate in the pulse width determination mode. Since rollovers were not of concern the timer control register was set up to mask timer interrupts. If one wishes to count rollovers a timer interrupt routine would have to be included in the monitor SLAVE firmware. The timer interrupt mask bit would then have to be cleared to insure response to timer interrupts.

MC146805E2 object code was generated by use of a Macro Cross Assembler. This code was then programmed into a 27C16 CMOS EPROM to produce the slave monitor.

A six volt rechargeable battery was used to power the circuitry since all devices were CMOS. Transmission back to the GIMIX serial card was understood in spite of the lack of a plus and minus supply at twelve volt levels. An important point to remember when building this circuit is to use a 47 microfarad filter capacitor as well as a couple of 0.1 microfarad bypass capacitors. If you don't want to fry eggs on your CMOS microprocessor. A battery isn't as continuous a source of electrical energy as one might think.

A small note of warning about the one shot circuit: this trick only works with CMOS Schmitt triggers. The bias currents at the input of any variety of TTL gate would charge up the capacitor and yield unpredictable results. Note, also, that there are two unused bits at the output port B of the processor; with a small effort one could decode these and monitor up to four pulse widths remotely. Alternately, these could initiate some response to a change of pulse width.

The source for the 307.2 kHz ceramic resonator is Radio Materials Company, 4242 N. Bryn Mawr Avenue, Chicago IL 60646. They have a \$150.00 minimum order.



A further feature of this circuit is that up to 128 other AART's can be used and addressed uniquely to ascertain pulse widths of various remote signals using the same receive, transmit and ground lines.

Jeffrey M. Cralg  
Apt. 912 - 3001 S. King Dr.  
Chicago, IL 60616

Christopher J. Ziolkowski  
3416 S. Hermitage  
Chicago, IL 60608

```

1.00=
2.00=* WHEN CROSS ASSEMBLYING CALL A MACRO
3.00= LIB MAC6805
4.00=
5.00=* THE NAME OF THIS PROGRAM IS SLAVE
6.00= NAM SLAVE
7.00= ORG $0001
8.00=
9.00=* EQUATES ARE LOCATIONS OF MC146805E2
10.00=* PORT AND TIMER REGISTERS
11.00=PDRA EQU $0000
12.00=PDRB EQU $0001
13.00=DDRA EQU $0004
14.00=ODRB EQU $0005
15.00=TDR EQU $0008
16.00=TCR EQU $0009
17.00=
18.00=* INITIALIZE PORT A
19.00=* ALL ARE OUTPUTS
20.00= LDA $FF
21.00= STA DDRA
22.00=* INITIALIZE PORT B
23.00=* SOME ARE INPUTS SOME ARE OUTPUTS
24.00= LDA #$11000010
25.00= STA DDRB
26.00=* GO INTO LOOP WAITING FOR DIVIDE BY NUMBER
27.00=LOOP1 NOP
28.00= BRCLR 5,PDRB,LOOP1
29.00=* GET DIVIDE BY NUMBER
30.00= LDA PDRB
31.00=* GET BITS INTO PROPER LOCATION
32.00= LSRA
33.00= LSRA
34.00=* ELIMINATE UNWANTED HIGH'S
35.00= AND #$00000111
36.00=* MAKE BIT PATTERN TO IGNORE TIMER INTERRUPTS
37.00=* MAKE BIT PATTERN FOR PULSE WIDTH MODE
38.00= ADD #$01010000
39.00= STA TCR
40.00=* START COUNTER AT 0
41.00= LDA $0
42.00= STA TDR
43.00=* CLEAR INTERRUPT MASK
44.00= CLI
45.00=* WAIT FOR INTERRUPT TO OCCUR
46.00=LOOP2 WAIT
47.00= BRA LOOP2
48.00=IRQ NOP
49.00= BRSET 0,PDRB,IRQ
50.00= LDA TDR
51.00= STA PDRA
52.00= LDA #$00000010
53.00= STA PDRB
54.00=* START COUNTER AT 0 AGAIN
55.00= LDA $0
56.00= STA TDR
57.00= RTI
58.00=

```

```

59.00= ORG $07FA
60.00= FCB $18
61.00= FCB $24
62.00= ORG $07FE
63.00= FCB $18
64.00= FCB $01

```

```

1.00=
2.00=* WHEN CROSS ASSEMBLYING CALL A MACRO
3.00= LIB MAC6805
4.00=
5.00=* THE NAME OF THIS PROGRAM IS SLAVE
6.00= NAM SLAVE
7.00= ORG $0001
8.00=
9.00=* EQUATES ARE LOCATIONS OF MC146805E2
10.00=* PORT AND TIMER REGISTERS
11.00=PDRA EQU $0000
12.00=PDRB EQU $0001
13.00=DDRA EQU $0004
14.00=ODRB EQU $0005
15.00=TDR EQU $0008
16.00=TCR EQU $0009
17.00=
18.00=* INITIALIZE PORT A
19.00=* ALL ARE OUTPUTS
20.00= LDAI $FF
21.00= STA DDRA
22.00=* INITIALIZE PORT B
23.00=* SOME ARE INPUTS SOME ARE OUTPUTS
24.00= LDAI $11000010
25.00= STA DDRB
26.00=* GO INTO LOOP WAITING FOR DIVIDE BY NUMBER
27.00=LOOP1 NOP
28.00= BRCLR 5,PDRB,LOOP1
29.00=* GET DIVIDE BY NUMBER
30.00= LDA PDRB
31.00=* GET BITS INTO PROPER LOCATION
32.00= LSRA
33.00= LSRA
34.00=* ELIMINATE UNWANTED HIGH'S
35.00= ANDI $00000111
36.00=* MAKE BIT PATTERN TO IGNORE TIMER INTERRUPTS
37.00=* MAKE BIT PATTERN FOR PULSE WIDTH MODE
38.00= ADDI $01010000
39.00= STA TCR
40.00=* START COUNTER AT 0
41.00= LDAI $0
42.00= STA TDR
43.00=* CLEAR INTERRUPT MASK
44.00= CLI
45.00=* WAIT FOR INTERRUPT TO OCCUR
46.00=LOOP2 WAIT
47.00= BRA LOOP2
48.00=IRQ NOP
49.00= BRSET 0,PDRB,IRQ
50.00= LDA TDR
51.00= STA PDRA
52.00= LDAI $00000010
53.00= STA PDRB
54.00=* START COUNTER AT 0 AGAIN
55.00= LDAI $0
56.00= STA TDR
57.00= RTI
58.00=
59.00= ORG $07FA
60.00= FCB $18
61.00= FCB $24
62.00= ORG $07FE
63.00= FCB $18
64.00= FCB $01

```

```

1.00=
2.00=* THE NAME OF THIS PROGRAM IS MASTER
3.00=* THE ROUTINES IN EQUATES ARE FLEX9
4.00=* THE SERIAL CARD IS ON PORT 4

```

```

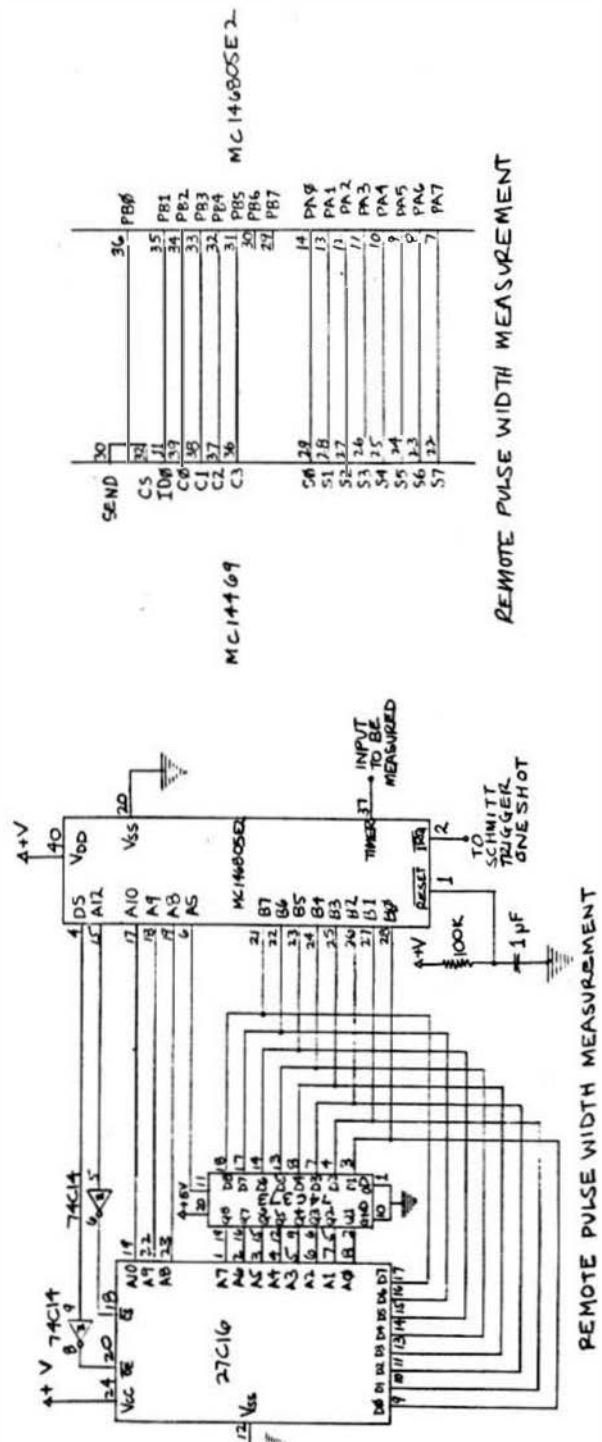
5.00= NAM MASTER
6.00= OPT PAG
7.00= ORG $1000
8.00=
9.00=DATA EQU $E011 THIS IS THE 6850'S DATA REGISTER
10.00=CONTRL EQU $E010 THIS IS THE 6850'S CONTRL REGISTER
11.00=PCRLF EQU $CD24
12.00=OUTHEX EQU $CD3C
13.00=WARMS EQU $CD03
14.00=OUTCH EQU $CDDF
15.00=MEM EQU $2000
16.00=
17.00=* CONFIGURE 6850 TO DIVIDE INCOMING
18.00=* DATA BY 16 CLOCK
19.00=* CONFIGURE 6850 FOR 8 BITS, EVEN PARITY
20.00=* ONE STOP BIT
21.00=BEGIN LDA A #$00011001
22.00= STA A CONTRL
23.00=* CHECK TO SEE IF TRANSMIT DATA REGISTER IS EMPTY
24.00= JSR LOOP1
25.00=* ADDRESS THE APPROPRIATE AART
26.00=CALL LDA A #$10000000
27.00= STA A DATA
28.00= JSR LOOP1
29.00=* SEND OUT A COMMAND WORD
30.00=* SELECT A DIVIDE BY ONE
31.00= LDA A #$00001000
32.00= STA A DATA
33.00= JSR LOOP1
34.00=* CHECK TO SEE IF DATA HAS BEEN RECEIVED
35.00=* FROM THE AART
36.00= JSR LOOP2
37.00=* GET DATA AND DISPLAY IT
38.00= LDA A DATA
39.00= STA A MEM
40.00= JSR PCRLF
41.00= LDX #MEM
42.00= JSR OUTHEX
43.00= LDA A #$20
44.00= JSR OUTCH
45.00= JSR LOOP2
46.00=* GET SECOND WORD (PULSE WIDTH) AND DISPLAY IT
47.00= LDA A DATA
48.00= STA A MEM
49.00= LDX #MEM
50.00= JSR OUTHEX
51.00= LDA A #$20
52.00= JSR OUTCH
53.00= JSR PCRLF
54.00=* LET'S HAVE A COKE BREAK (DELAY)
55.00= LDX #$FFFF
56.00=LOOP3 DEX
57.00= CPX #$0000
58.00= BNE LOOP3
59.00=* LET'S GET ANOTHER TWO WORDS LIKE ABOVE
60.00= JSR LOOP1
61.00=CALL1 LDA A #$10000000
62.00= STA A DATA
63.00= JSR LOOP1
64.00= LDA A #$00001000
65.00= STA A DATA
66.00= JSR LOOP1
67.00= JSR LOOP2
68.00= LDA A DATA
69.00= STA A MEM
70.00= JSR PCRLF
71.00= LDX #MEM
72.00= JSR OUTHEX
73.00= LDA A #$20
74.00= JSR OUTCH
75.00= JSR LOOP2
76.00= LDA A DATA
77.00= STA A MEM
78.00= LDX #MEM
79.00= JSR OUTHEX
80.00= LDA A #$20
81.00= JSR OUTCH
82.00= JSR PCRLF
83.00=* LET'S GO BACK TO FLEX9 DOS
84.00= JMP WARMS

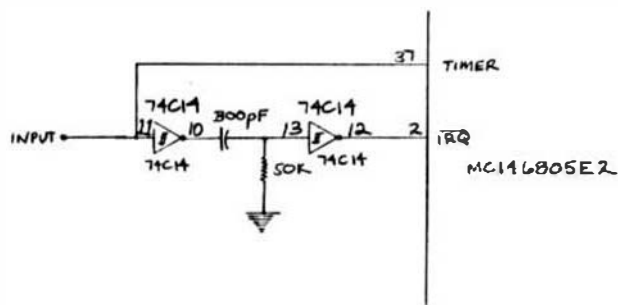
```

```

85.00=LOOP1 LDA A CONTRL
86.00= AND A #$00000010
87.00= CMP A #$00000010
88.00= BNE LOOP1
89.00= RTS
90.00=LOOP2 LDA A CONTRL
91.00= AND A #$00000001
92.00= CMP A #$00000001
93.00= BNE LOOP2
94.00= RTS
95.00= END BEGIN

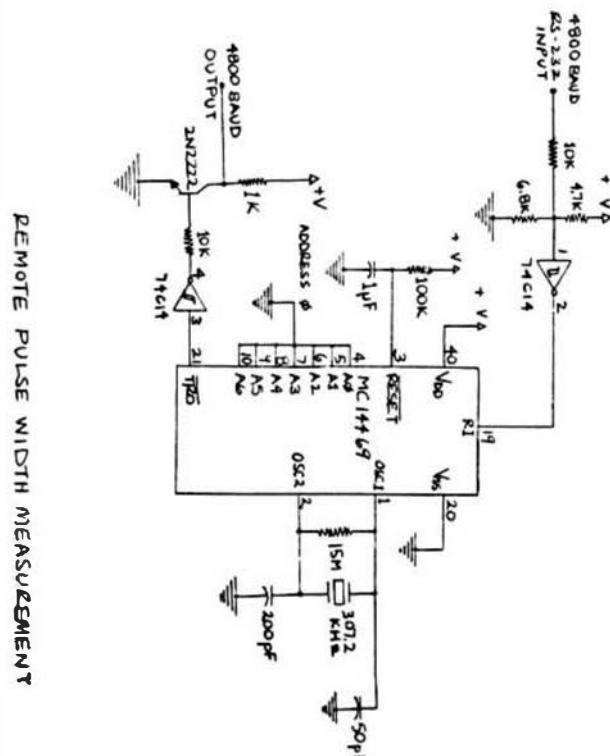
```



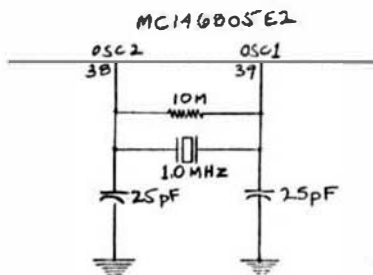


SCHMITT TRIGGER ONE SHOT

REMOTE PULSE WIDTH MEASUREMENT



REMOTE PULSE WIDTH MEASUREMENT



SYSTEM CLOCK

REMOTE PULSE WIDTH MEASUREMENT

## James Bellomo Dies of Heart Attack

James Bellomo of JOTO ASSOCIATES died February 25, 1984 of a heart attack. Jim was an active SS-50 Bus supporter, and was the "OS-9 Special Interest Group" system operator on Compuserve. He is survived by his wife Mako and 12 year old son David. At 34 years of age it seems that his family and our industry were cheated of a truly dedicated and committed individual. Our prayers go out to his family at this time. Associates will continue to serve those who did business with Jim.

## DAY COMMAND FOR FLEX-9

Dear Larry:

I am enclosing a Flex-9 disk containing an article for '68' Micro, entitled "DAY Command for Flex-9" and the source and command files for this utility. I really like DAY.CMD and I trust your readers will benefit from it. Maybe you should have a "utility of the month club"?

Yours Truly

J. Gary Mills

Ed's Note: Thanks Gary, sounds like a great idea, now how about the rest of you? Send in your favorite utility (non-copyrighted) and win subscription renewals, books and/or world fame! DMW

### DAY Command for Flex-9

J. Gary Mills  
1019 Weatherdon Ave.  
Winnipeg, Manitoba  
Canada R3M 2B5

Here is a utility for Flex-9 that makes your system a little more friendly. The DAY command simply displays the current day of the week plus the date, month, and year.

An example is: "Sunday, 29 January 1984".

It is an alternative to the DATE command.

I use it as a part of my startup file to confirm that I have entered the correct date. I also use it to annotate printer listings, for example: "P DAY:P LIST filename".

Each element of the display is produced by a separate subroutine. This makes it easy to change the format by re-arranging the LBSR statements. Flex users in the USA may wish, for example, to have it say: "Sunday, January 29, 1984".

Although the utility runs in the Utility Command Area, it is written in position-independent code. It examines the Flex date register and calculates the day of the week. This requires some "astronomical" calculations, including 3-byte arithmetic and floating point simulation by

fractions. The algorithm is explained in the comments of the DAYPRT routine. This routine calls routines that emulate a CPU with a 24-bit accumulator. The result of the calculation is a number between zero and six, corresponding to a day between Sunday and Saturday. The address of the string which is printed is obtained from an index that precedes the group of strings.

- - -

\*  
\* "DAY" COMMAND  
\*

\* AUTHOR: J. GARY MILLS  
\* 1019 WEATHERDOWN AVE  
\* WINNIPEG, MAN  
\* CANADA R3M 2B5

\* FLEI ADDRESSES

CD03	WARM5	EDU	#CD03	FLEI RETURN POINT
CD18	PUTCHR	EDU	#CD18	PUT A CHAR TO CONSOLE
CD24	PCRLF	EDU	#CD24	START NEW LINE ON CONSOLE
CD39	OUTDEC	EDU	#CD39	PRINT A DECIMAL NUMBER
CC0E	SYDR	EDU	#CC0E	SYSTEM DATE REGISTER
CC0E	SYDRM	EDU	SYDR+0	, MONTH
CC0F	SYDRD	EDU	SYDR+1	, DAY
CC10	SYDRY	EDU	SYDR+2	, YEAR

C100		ORG	#C100	UCA
C100 20	16	DAYCMD	BKA	DISPLA
C102 01		FCB	1	VERSION #

\* VARIABLES

C103	ACC	RMB	3	ACCUMULATOR
C106	DAY	RMB	3	DAY
C109	MON	RMB	3	MONTH
C10C	YEA	RMB	3	YEAR

\* CONSTANTS

C10F 00	K1	FCB	0	365.25
C110 016D		FDB	365	
C112 00	K2	FCB	0	30.6
C113 001E		FDB	30	
C115 F6 06 07	K3	FCB	0F6,0B6,007	-621049

\* PRINT FORMATTED DAY AND DATE

C118	DISPLA	EDU	*	
C118 0D	CD24	JSR	PCRLF	START NEW LINE
C118 17	0015	LBSR	DAYPRT	PRINT DAY
C11E 17	013D	LBSR	CONSPA	PRINT COMMA AND SPACE
C121 17	0141	LBSR	DATPRT	PRINT DATE
C124 17	0139	LBSR	SPACE	PRINT A SPACE
C127 17	0149	LBSR	MONPRT	PRINT MONTH
C12A 17	0133	LBSR	SPACE	PRINT A SPACE
C12D 17	0151	LBSR	YEAPRT	PRINT YEAR
C130 7E	CD03	JMP	WARM5	GOTO FLEI

\* PRINT THE CURRENT DAY OF THE WEEK

C133	DAYPRT	EDU	*	
C133 CC	0000	LDB	00	
C136 ED	0C CD	STD	DAY,PCR	
C139 ED	0C CD	STD	MON,PCR	
C13C ED	0C CD	STD	YEA,PCR	
C13F F6	CC0E	LDB	SYDRM	

C142 C1	02	CMPB	#2	IF MONTH > 2
C144 23	0F	BLS	DA07	
C146 3C		INCB		
C147 E7	0C C1	STB	MON+2,PCR	MONTH := MONTH + 1
C14A F6	CC10	LDB	SYDRY	
C14D C3	076C	ADDD	#1900	
C150 E0	0C 0A	STD	YEA+1,PCR	
C153 20	0E	BRA	DA12	ELSE
C155 C8	0D	ADDB	#13	
C157 E7	0C 01	STB	MON+2,PCR	MONTH := MONTH + 13
C15A F6	CC10	LDB	SYDRY	
C15D C3	076D	ADDD	#1899	YEAR := YEAR - 1
C160 E0	0C AA	STD	YEA+1,PCR	
C163 B6	CC0F	LDA	SYDRD	
C166 A7	0C 9F	STA	DAY+2,PCR	
C169 31	0C A3	LEAY	K1,PCR	
C16C 0D	3D	BSR	LOAD	
C16E 31	0C 9B	LEAY	YEA,PCR	ACC := YEAR * 365.25
C171 17	00B4	LBSR	MULT	
C174 C6	02	LDB	#2	
C176 BD	62	BSR	SRAD	
C178 31	0C 00	LEAY	DAY,PCR	
C17B 0D	48	BSR	ADDD	DAY := DAY + ACC
C17D 0D	39	BSR	STOR	
C17F 31	0C 90	LEAY	K2,PCR	
C182 0D	27	BSR	LOAD	
C184 31	0C 02	LEAY	MON,PCR	ACC := MONTH * 30.6
C187 0D	6F	BSR	MULT	
C189 C6	01	LDB	#1	
C18B 0D	4D	BSR	SRAD	
C18D C6	03	LDB	#3	
C18F 0D	49	BSR	SRAD	
C191 31	0C 01	LEAY	K3,PCR	
C194 0D	2F	BSR	ADDD	
C196 31	0D FF6C	LEAY	DAY,PCR	
C19A 0D	29	BSR	ADDD	DAY := DAY + ACC - 621049
C19C 0D	1A	BSR	STOR	
C19E 17	0090	LBSR	REMR	CALC MODULUS 7
C1A1 30	0D 00E6	LEAY	DINDEX,PCR	--> INDEX
C1A5 E6	86	LDB	A,I	LOOKUP
C1A7 3A		AB1		--> NAME
C1AB 16	00A5	LBRA	PDATA	PRINT IT

\* LOAD 3-BYTE ACCUMULATOR

* Y --> SOURCE	
C1AB	LOAD EDU *
C1AD EC	21 LDB 1,Y
C1AD ED	0D FF53 STD ACC+1,PCR
C181 A6	A4 LDA 0,Y
C183 A7	0D FF4C STA ACC,PCR
C187 39	RTS

\* STORE 3-BYTE ACCUMULATOR

* Y --> DESTINATION	
C188	STOR EDU *
C188 EC	0D FF4B LDB ACC+1,PCR
C18C ED	21 STD 1,Y
C18E A6	0D FF41 LDA ACC,PCR
C1C2 A7	A4 STA 0,Y
C1C4 39	RTS

\* ADD 3-BYTE NUMBER TO ACCUMULATOR

* Y --> NUMBER	
C1C5	ADDD EDU *
C1C5 EC	21 LDB 1,Y
C1C7 E3	0D FF39 ADDB ACC+1,PCR
C1C8 ED	0D FF35 STD ACC+1,PCR
C1CF A6	A4 LDA 0,Y
C1D1 A9	0D FF2E ADCA ACC,PCR
C1D5 A7	0D FF2A STA ACC,PCR
C1D9 39	RTS

## FOR THE ELEKTRA

- **OS-9** with Editor, Assembler, and Debugger **\$250.00**
- **STAR-DOS Level 1** (FLEX™ compatible but with up to 10 active drives:  
i.e. 4 8" floppies, 4 5" floppies, and 2 Winchester drives)  
Regular price **\$75.00**      Introductory price **\$50.00**

## FOR OS-9 by Epstein Associates

- **Super Modem Program** with autodial, configuration file, etc.  
(Available exclusively through AAA Chicago Computer Center) **\$100.00**
- Coming — Complete software for Western Union EasyLink

## THE ELEKTRA SUPER FLOPPY CONTROLLER

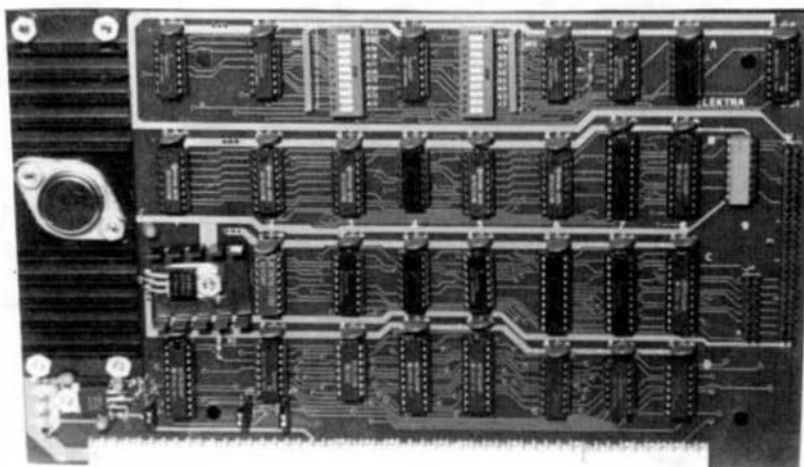
- Emulates the DC-1, DC-2, DC-3, DC-4 as well as the GIMIX #28, #38, #48, and #58 controllers **\$275.00**

## THE ELEKTRA WINCHESTER SYSTEMS with up to 11 bit burst error correction

- |            |   |                  |                               |                  |
|------------|---|------------------|-------------------------------|------------------|
| <b>New</b> | 7 Megabyte Single Drive System                      | <b>\$1995.00</b> | 14 Megabyte Dual Drive System | <b>\$2995.00</b> |
| •          | 12 Megabyte Single Drive System                     | <b>\$2295.00</b> | 24 Megabyte Dual Drive System | <b>\$3595.00</b> |
| •          | 19 Megabyte Single Drive System                     | <b>\$2995.00</b> | 38 Megabyte Dual Drive System | <b>\$4695.00</b> |
| <b>New</b> | Winchester Electronics (no drives)                  |                  |                               | <b>\$995.00</b>  |
| <b>New</b> | DMA SS-50C Bus Interface (No controller nor drives) |                  |                               | <b>\$695.00</b>  |

## THE ELEKTRA DBI BOARD

(DMA SS-50C Bus Interface for the ELEKTRA Winchester Systems)



Phone:

**AAA Chicago Computer Center**

Technical Consultation available most weekdays from 4 p.m. to 6 p.m. CST

(312) 459-0450

120 Chestnut Lane

Wheeling, IL 60090

See our catalog and ordering information on the next page.



# HELIX

64K 6809 Computer	\$2395.00	64K 68008 Computer	\$2495.00
256K 6809 Computer	2895.00	256K 68008 Computer	2995.00
Other systems available			
20 Megabyte 5" add on Winchester System			\$2595.00
64K CMOS Static memory board with battery backup			395.00
DMA 5" and 8" Floppy Controller with built in Winchester controller I/O			695.00
DMA 5" and 8" Floppy Controller with built in 6809 CPU Board			495.00
68008 board for SS-50	595.00	CP/M-68K	350.00

Need FLEX, UniFLEX, OS-9 Level 1, or OS-9 Level II? We have a system for you!

**ELEKTRA COMPUTER CABINET THE LARGEST SS-50 COMPUTER CABINET AVAILABLE!** Made of heavyweight 0.090" thick aluminum. Interior is 18-1/2" wide by 21-7/8" deep by 8-3/4" high. Heavy duty A.C. line cord. A.C. fuse holder. EMI filter. Fan with filter. Back panel has 10 cutouts for 'D' type data connectors. Front panel has key on/off power switch, 2 illuminated push button switches (Reset and NMI/Abort) and two cutouts for 5-1/4" disk drives.

Filter Plate for 5-1/4" drive opening: \$10.00 Fan Filter: \$10.00

**POWER SUPPLY** Highest quality linear power supply CONSERVAT: ELY rated at 15a @ 5v, 3a @ 10v, 3a @ 15v. Multi-tapped primary for line tuning \$200.00

**DISK REGULATOR BOARD WITH CABLES** Standard version for 2 floppy drives \$50.00 Heavy duty version for 1 Winchester drive and 1 floppy drive \$75.00

**AUXILIARY POWER SUPPLY** to power second Winchester drive \$125.00

**ELEKTRA UNIVERSAL 68-6809-68C SOFTWARE BOARD** 0.125" thick 16" long by 9" wide 11 memory (50 pin) slots 8 I/O (30 pin) slots. Complete address decoding and bus control as well as external address expansion, for I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. 1" spacing between all memory and I/O slots. On board baud rate generator with low and high ranges providing jumper selectable rates of 75 through 38,400 for each of the five baud rate lines. Also device circuitry permitting 1 Mhz 30 pin disk controllers to run with 2MHz 50 pin CPU boards.

Mounting hardware \$5.00 Bareboard w/documentation: \$80.00 Kit w/gold connectors: \$400.00 Assembled w/gold connectors: \$460.00 Kit with connectors: \$320.00 Assembled with connectors: \$380.00

**ELEKTRA CHASSIS** includes cabinet, 110v power supply, power supply cables, standard disk regulator board with power cables, motherboard with gold square pin connectors, assembled and tested. \$950.00

**ELEKTRA CPU 8/9** Use either the 6802 or 6808 (to run 6800 software) or 6809. Has provision for up to 3 37718 Eproms. 1K scratchpad. MC6840 triple timer, and an optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. Run O-9 FLEX™, STAR-DOS Bareboard: \$50.00 Kit: \$225.00 Assembled: \$275.00 Optional Baud Rate Generator: \$25.00

**ELEKTRA OPS DUAL PORT SERIAL CARD** Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per port. RTS, CTS, DTR, DCD, I/O, FIRQ/NMI, and baud rate can be appropriately implemented for each port. Bareboard: \$30.00 Kit: \$75.00 Assembled: \$85.00 Cable (two needed per board): Each: \$20.00

**ELEKTRA DPP DUAL PORT PARALLEL CARD** Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request line for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line. Bareboard: \$20.00 Kit: \$60.00 Assembled: \$80.00 Cable (two needed per board): Each: \$20.00

**ELEKTRA 64K STATIC RAM/ROM MEMORY BOARDS** with gold connectors (in available) Assembled and tested. With 56K RAM \$269.00 With 64K RAM \$299.00 Kit With 56K RAM \$219.00 With 64K RAM \$249.00

**ELEKTRA UNIVERSAL SUPER FLOPPY CONTROLLER THE BEST 30 PIN FLOPPY DISK CONTROLLER THAT YOU CAN BUY!** Controls up to four 5-1/4" drives and four 8" drives for a total of eight system drives. (FLEX system limit is four drives.) Single density or double density, 1MHz or 2MHz, 6800 or 6809 (Double density 8" must be at 2MHz, all other combinations of performance are possible.) Analog phase locked loop data separators with separate adjustments for 5" and 8" drives. Analog write precompensation circuit with separate adjustments for 5" and 8" drives. Designed to meet the data hold requirements of Western Digital floppy controller IC. Assembled and tested. \$275.00

Disk with drivers, setup, and formatting utilities. Specify 6800 (FLEX2 or Gen. FLEX) or 6809 (FLEX9 or Gen. FLEX), or OS-9, 5" or 8". \$30.00 STAR-DOS (Specify 5" or 8"). Regular price \$75.00 Introductory price \$50.00

**ELEKTRA WINCHESTER SYSTEMS THE BEST WINCHESTER SYSTEMS THAT YOU CAN BUY!** Has automatic error detection and CORRECTION of up to 11 bit burst errors. SS-50 bus, extended addressing capabilities. DMA, on board sector buffer, drivers included for 6800 FLEX or OS-9. Specify whose version of FLEX that you are using. Drivers for FLEX2 (6800) are available for an additional \$100.00. Price includes host (interface, controller, drivers), and cables.

7 Megabyte single drive sys. \$1995.00 14 Megabyte dual drive sys. \$2995.00 12 Megabyte single drive sys. \$2295.00 24 Megabyte dual drive sys. \$3595.00 19 Megabyte single drive sys. \$2995.00 38 Megabyte dual drive sys. \$4695.00 (10 Megabyte drives are the largest that can be supported by FLEX)

Circuit boards, cables, software (No drives) \$95.00 SS-50C DMA Bus Interface board only \$65.00

**ELEKTRA HD-5** Cabinet for dual 5-1/4" floppy drives with power supply, line cord, fuse power switch, and power cables to drives \$150.00

**ELEKTRA HD-8W** As above but with EMI filter, fan, and heavy duty power supply. Powers 1 floppy and 1 Winchester. \$199.00

5" ribbon cable for dual outboard 5-1/4" disk drives \$40.00 2" ribbon cable for dual inboard 5-1/4" disk drives \$35.00 Custom cables available Phone

**ELEKTRA HD-8 Dual 8" drive cabinet, EMI filter, fan with filter, power supply and power supply cables.** \$350.00

8" ribbon cable for dual 8" disk drives \$45.00

**ELEKTRA 30 PIN PROTOTYPING BOARD** \$20.00 **ELEKTRA 60 PIN PROTOTYPING BOARD** \$40.00

**GOLD 10 PIN CONNECTORS** (Specify male with square pins or female) \$1.50 **TIN 10 PIN CONNECTORS** (Specify male with square pins or female) \$0.50

**ELEKTRA** is a trademark of AAA Chicago Computer Center. **FLEX** and **UniFLEX** are trademarks of Technical Systems Consultants, Inc. **HELIX** is a trademark of Hazeltown Computer Systems. **OS-9** and **BASIC09** are trademarks of Microware Systems Corp.

**AAA CHICAGO COMPUTER CENTER** (312) 459-0450 120 CHESTNUT LANE • WHEELING, IL 60090 Technical consultation available 4 PM to 6 PM most weekdays. Closed evenings and weekends.

**TERMS** Minimum order \$20.00. Shipping and handling estimates within the Continental U.S., add 3% (MINIMUM \$2.50). Illinois residents add 6% sales tax. We will refund your overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINIMUM \$10.00). Foreign orders must be prepaid in U.S. dollars. Checks must be drawn on a U.S. bank. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 6 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa, and American Express honored.

Our apology: We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The too frequent changing of our inventory and prices makes it unpractical to publish a catalog. Our ads are intended to serve that purpose. Prices and inventory are subject to change without advance notice.

**ELEKTRA™ SOFTWARE** (All of our software is copyrighted and all rights are reserved. Source is either supplied or optionally available at extra cost so that the purchaser can modify our programs for his own use. Licensing, however, is required for commercial resale.)

**SUPER MODEM PROGRAM** Single character commands. No interrupts required. Transmit manually or transmit disk files (text) of any length to distant computer. Receive and save disk files (text) on local disk system. X-on/X-off supported. Tested for full duplex at speeds up to 9600 baud. Half duplex option. Echo option. Replaces CR with CR/LF (user option). Slow disk file transmit option.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8". \$75.00 Instruction Manual and disk with both source and object code

**STANDARD MODEM PROGRAM** Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Specify 6800 or 6809.

Manual with instructions, source listing, and flow chart \$45.00

**OS-9 Configurable Modem Program** (Sorry, source is not available) \$100.00

**ORDER — WRITE UP COMPUTER PROGRAM** Screen oriented write up form with cursor editing, disk save and load, printer command using easily available universal print-out forms. Phone for more details. Available for 6809 FLEX. \$100.00

**ALL IN ONE** Editor — Text Processor — Mailing Labels — Mailing Lists — Multiple Form Letters Use any CRT terminal and printer — Best Package For The Money Anywhere!

Specify 6800 or 6809, SSB or FLEX™, 5" or 8". \$75.00 Add \$35.00 for printed source listing; add \$100 for source on disk.

All-In-One, Write'n Spell, and Spell'n Fix package \$250.00

**Software by Technical Systems Consultants, Inc.**

	Adapt. Guide	Source (List)	FLEX™ Source (Disk)	Man. Only	Object w/Man.	Add. Man.	Man. Only	Object w/Man.
Gen. DOS w/Edit & ASMB	—	—	—	50	250	—	—	—
SWTPC DOS w/Edit & ASMB	—	—	—	25	150	40	100	550
Advanced Programmers Guide	—	—	—	25	—	—	—	—
Editor	100	250	25	50	—	—	—	—
Assembler	150	250	25	50	—	—	—	—
Debugger	175	250	25	75	—	—	—	—
Extended Basic	—	—	—	25	100	20	50	200
Basic Precompiler	—	—	—	25	50	10	25	150
Sort/Merge	—	—	—	25	75	20	35	150
Utilities	—	Inc.	25	75	10	25	150	—
Diagnostics	—	—	—	25	75	—	—	—
Text Processor	150	250	25	75	20	35	150	—
68000 X-ASMB on 6809	—	—	—	25	250	20	35	300
Rel ASMB/Linking Loader	—	—	—	50	200	25	50	300
6800 X-ASMB on 6809	—	—	—	25	150	20	35	175
Cobol	—	—	—	—	100	—	—	—
Fortran 77	—	—	—	—	30	75	750	—
					35	65	450	—

**Software by Microware Systems Corp.** (Suggested List Prices, varies w/mfg)

	Run-Time Package	Source	Manual Only	Object Only
OS-9™ Level 1 w/Edit, Asm, Debug	—	400.00	40.00	250.00
OS-9™ Level 2 w/Edit, Asm, Debug	—	400.00	40.00	300.00
OS-9™ Edit, Asm, Debug Pkg	—	—	25.00	125.00
Device Driver for Disk Controller (Specify Model)	100.00	—	—	—
Device Driver for ACIA and PIA	50.00	—	—	—
Clock Driver for 6840 and 58167 clock chips	35.00	—	—	—
Entertainment Pack 1, or File Handler Toolbox, or NineCom	—	—	10.00	85.00
Print Spooler (Level 2 only)	—	—	15.00	150.00
Virtual Disk Driver (Level 2 only)	—	—	10.00	85.00
RMA Relocatable Macro Assembler	—	—	20.00	200.00
RMA/68000 Cross Assembler	—	—	—	400.00
BASIC09™ W/Run-Time	50.00	N/A	25.00	200.00
BASIC09 Tour Guide Book	—	—	18.95	—
C-Compiler	—	—	25.00	—
C Programming Language (Kernighan & Ritchie)	—	—	19.95	250.00
CIS Cobol Compiler w/Forms 2 Prog. Gen.	50.00	N/A	40.00	400.00
Pascal Compiler	50.00	N/A	25.00	250.00
Sage Application Generator	300.00	N/A	25.00	995.00
Microware yearly support service (All products)	—	—	—	150.00
Edition Update w/manuals	25.00	—	—	75.00

**Special Software** STAR-DOS Level 1 Regular Price \$75.00 Introductory Price \$50.00

2K 6809 MICROBUG 40.00 4K 6809 HUMBUB 75.00 2K 6800 HUMBUB 40.00

Other HUMBUB versions including video versions are available. \$75.11

Spell'n Fix by Peter Stark 178.58 Write'n Spell by Peter Stark 250.00

All-In-One, Spell'n Fix, and Write'n Spell package \$99.00

**SUPER SLEUTH Disassembler System** (\$101.00 for OS-9 version)

**SD/DD DISK DRIVES** 1 head 2 heads 2 heads 1 head 2 heads

30 day guarantee Tandon Tandon CDC MPI MPI

5-1/4" 40 tracks 225.00 300.00 300.00 250.00 325.00

5-1/4" 80 tracks 300.00 375.00 375.00 325.00 400.00

MPI or CDC Service Manual (Specify 40 or 80 tracks) 25.00

8" 77 tracks SD/DD Qume DT-8 \$550.00 Remex (Special) 350.00

**SPECIALS** U.S. Robotics 300/1200 baud auto dial/auto answer modem \$99.00

\* Same as above but without self test and diagnostics \$99.00

\* U.S. Robotics 1200 baud direct connect auto answer modem \$299.00

\* SSB BFD Floppy Disk Controller (Version 3) Run FLEX or SSB DOS \$100.00

\* SWTPC 4K Memory \$15.00 MP-Mb (4K bareboard) 9.95

\* High speed tape reader 50.00 300 Baud acoustic modem \$129.00

\* TI 810 Printer w/lower case and full vertical forms control \$1200.00

\* Microtime II Calendar and Clock Board Assembled 60.00 Bare 30.00

**GIMIX CLEARANCE SALE** LIST OUR PRICE LIST OUR PRICE

#5 6809 Plus CPU Bd. 578.05 475.00 6800 CPU board 224.03 100.00

Cable (Ser or Par I/O) 24.95 20.00 8 Port Serial I/O Bd. 318.46 250.00

Double disk reg. card 68.22 50.00 #28 control w/GMX Flex 328.28 270.00

80 X 24 Video Boards 398.76 250.00 Single prt ser. 1 cable 113.36 90.00

```

* RIGHT-SHIFT 2-BYTE NUMBER & ADD TO ACCUMULATOR
* Y --> NUMBER
* 8 = BIT COUNT
C10A 34 04 C1DA SRAD EQU #
C10C EC 21 PSWS 8
C10E 44 SR01 LDD 1,Y
C10F 56 RORB
C1E0 6A E4 DEC 0,S
C1E2 26 FA BNE SR01
C1E4 E3 8D FF1C ADDD ACC+1,PCR
C1E8 ED 8D FF18 STD ACC+1,PCR
C1EC A6 8D FF13 LDA ACC,PCR
C1F0 89 00 ADCA 00
C1F2 A7 8D FF0D STA ACC,PCR
C1F6 35 84 PULS 0,PC

```

```

* MULTIPLY TWO 2-BYTE NUMBERS
* Y --> NUMBER ACC = NUMBER
* ACC = RESULT
C1F8 32 7D C1FB MULT EQU #
C1FA 6F E4 LEAS -3,S
C1FC A6 8D FF05 CLR 0,S
C200 E6 22 LDA ACC+2,PCR
C202 38 22 LDB 2,Y
C203 E8 61 MUL
C205 A6 8D FEF3 STD 1,S
C209 E6 22 LDA ACC+1,PCR
C20B 3D 22 LDB 2,Y
C20C E3 E4 MUL
C20E ED E4 ADDD 0,S
C210 A6 8D FEF1 STD 0,S
C214 E6 21 LDA ACC+2,PCR
C216 3D 21 LDB 1,Y
C217 E3 E4 MUL
C219 ED E4 ADDD 0,S
C21B A6 8D FEE5 STD 0,S
C21F E6 21 LDA ACC+1,PCR
C221 3D 21 LDB 1,Y
C222 ED E4 MUL
C224 E7 8D FED8 ADDD 0,S
C228 EC 41 STD ACC,PCR
C22A ED 8D FED6 LDB 1,S
C22E 32 63 STD ACC+1,PCR
C230 39 LEAS 3,0
RTS

```

```

* CALC REMAINDER ON DIVISION BY SEVEN
* Y --> 3-BYTE NUMBER
* A = REMAINDER
C231 E6 10 C23I REMDR EQU #
C233 34 T4 LDB 016
C235 EC A4 PSWS 6
C237 68 22 LDB 0,Y
C239 59 RE01 LSL 2,Y
C23A 49 ROLB
C23B 24 04 BOLA
C23D 88 07 BCC RE02
C23F 20 04 ADDA 07
C241 80 07 BRA RE03
C243 6C 22 SUBA 07
C245 6A E4 INC 2,Y
C247 26 EE DEC 0,S
C249 40 BNE RE01
C24A 2A 02 TSTA
C24C 88 07 BPL RE04
C24E 35 84 ADDA 07
PULS 0,PC

```

```

* PRINT A STRING
* X --> STRING

```

```

C250 A6 80 PDATA EQU #
C252 81 04 PD01 LDB 1,X+
C254 27 05 CMPA 0004
C256 BD C018 BEQ PD02
C259 20 F5 JSR PUTCHR
C25B 39 PD02 BRA PD01
RTS

```

```

* PRINT A COMMA AND A SPACE

```

```

C25C 86 2C C25C COMSPA EQU #
C25E 8D 02 LDA 0',
BSR SP01

```

```

* FALL INTO SPACE

```

```

* PRINT A SPACE

```

```

C260 86 20 C260 SPACE EQU #
C262 7E C018 LDA 0+20
SP01 JMP PUTCHR

```

```

* PRINT THE CURRENT DATE

```

```

C265 4F C265 DATPAT EQU #
C266 F6 CCOF CLR A
C269 30 8D FE97 D401 LDB SYDRD
C26B ED 84 LEA/ ACC+1,PCR
C26F 5F 84 STD 0,X
CLR B
C270 7E ED39 JMP OUTDEC

```

```

* PRINT THE CURRENT MONTH

```

```

C273 86 C273 MONPRT EQU #
C276 4A CCOE LDA SYDRM
DECA
C277 30 8D 0050 LEA/ MINDEX,PCR --> INDEX
C278 E6 86 LDB A,X LOOKUP
C27D 3A AB/ --> NAME
C27E 16 FFDF LBRA PDATA PRINT 17

```

```

* PRINT THE CURRENT YEAR

```

```

C281 4F C281 YEAPRT EQU #
C282 F6 CCI0 CLR A
C285 C3 076C LDB SYDRY
C288 16 FFDE ADDD 01900
LBRA D401

```

```

* INDEX FOR DAY STRINGS

```

```

C28B 07 C28B DINDEX EQU #
C28C 0E FCB DASUN-DINDEX
C28D 15 FCB DAMON-DINDEX
C28E 18 FCB DATUE-DINDEX
C28F 27 FCB DAMED-DINDEX
C290 30 FCB DATMU-DINDEX
C291 37 FCB DAFRI-DINDEX
FCB DASAT-DINDEX

```

```

* DAY STRINGS

```

```

C292 53 75 6E 64 DASIM FCC "Sunday"
C296 61 79 FCB 004
C298 04 FCB 004
C299 40 6F 6E 64 DAMON FCC "Monday"
C29D 61 79 FCB 004
C29F 04 FCB 004
C2A0 54 75 65 73 DATUE FCC "Tuesday"
C2A4 64 61 79 FCB 004
C2A7 04 FCB 004
C2A8 57 65 64 6E DAMED FCC "Wednesday"
C2AC 65 73 64 61

```

C2B0 79		
C2B1 04		
C2B2 54 68 75 72	DATHU	FCC \$04 "Thursday"
C2B6 73 64 61 79		
C2BA 04		
C2B8 46 72 69 64	DAFRI	FCC \$04 "Friday"
C2BF 61 79		
C2C1 04		
C2C2 53 61 74 75	DASAT	FCC \$04 "Saturday"
C2C6 72 64 61 79		
C2CA 04		

DATHU C2B2	DATPRT C265	GATUE C2A0	DAMED C2AB	DAY C106
DAYCMD C100	DAYPRT C133	DINDEX C2BB	DISPLA C11B	RI C10F
K2 C112	K3 C115	LOAD C1AB	MINDEX C2CB	MOAPR C2EE
MOAUG C302	MODEC C324	MOFEB C20F	MOJAN C207	MOJUL C2FD
MOJUN C2F8	MONAR C2E8	MONAY C2F4	MON C109	MONOV C31B
MONPRT C273	MOOCT C313	MOSEP C309	MULT C1FB	OUTDEC C039
PCRLF C024	P001 C250	P002 C25B	PDATA C2S0	PUTCHR C01B
RE01 C237	RE02 C241	RE03 C245	RE04 C24E	REMR C231
SP01 C262	SPACE C260	SR01 C1DE	SRAD C1DA	STR C1BB
SYDR C00E	SYDRD C00F	SYORM C00E	SYDRY C010	WARMS C003
YEA C10C	YEAPRT C2B1			

\* INDEX FOR MONTH STRINGS

C2CB MINDEX	EQU *
C2CB 0C	FCB MOJAN-MINDEX
C2CC 14	FCB MOFEB-MINDEX
C2CD 10	FCB MONAR-MINDEX
C2CE 23	FCB MOAPR-MINDEX
C2CF 29	FCB MONAY-MINDEX
C2D0 20	FCB MOJUN-MINDEX
C2D1 32	FCB MOJUL-MINDEX
C2D2 37	FCB MOAUG-MINDEX
C2D3 3E	FCB MOSEP-MINDEX
C2D4 4B	FCB MOOCT-MINDEX
C2D5 50	FCB MONOV-MINDEX
C2D6 59	FCB MODEC-MINDEX

\* MONTH STRINGS

C2D7 4A 61 6E 75	MOJAN	FCC "January"
C2D8 61 72 79		
C2DE 04		
C2DF 46 65 62 72	MOFEB	FCC \$04 "February"
C2E3 75 61 72 79		
C2E7 04		
C2E8 40 61 72 63	MONAR	FCC \$04 "March"
C2EC 6B		
C2ED 04		
C2EE 41 70 72 69	MOAPR	FCC \$04 "April"
C2F2 6C		
C2F3 04		
C2F4 4D 61 79	MONAY	FCC \$04 "May"
C2F7 04		
C2F8 4A 75 6E 65	MOJUN	FCC \$04 "June"
C2FC 04		
C2FD 4A 75 6C 79	MOJUL	FCC \$04 "July"
C301 04		
C302 41 75 67 75	MOAUG	FCC \$04 "August"
C306 73 74		
C308 04		
C309 53 65 70 74	MOSEP	FCC \$04 "September"
C30D 65 60 62 65		
C311 72		
C312 04		
C313 4F 63 74 6F	MOOCT	FCC \$04 "October"
C317 62 65 72		
C31A 04		
C31B 4E 6F 76 65	MONOV	FCC \$04 "November"
C31F 6B 62 65 72		
C323 04		
C324 4A 65 63 65	MODEC	FCC \$04 "December"
C32B 6D 62 65 72		
C32C 04		

END DAYCMD

0 ERROR(S) DETECTED

SYMBOL TABLE:

ACC	C103	ADD3	C1C5	COMSPA	C25C	DA01	C269	DA07	C155
DA12	C163	DAFRI	C2BB	DAMON	C299	DASAT	C2C2	DASUN	C292

# BIT BUCKET

## OS-9 USER GROUP NEWS

### CONTINUED

#### OS-9 Users Group Abroad

Our communications points for Europe and Japan respectively will be:

Dr. Rudolf Keil  
Keil Software-Elektronik-Datentechnik  
Porphyrtstrasse 15  
D-6905 Schriesheim  
Tel.: (06203) 6741  
Telex: 4 65 025 keil d

Schriesheim is near Heidelberg.

Toshio Shinjo  
Microboards, Inc.  
3-8-9 Baraki  
Ichikawa City, Chiba 272-01  
Japan  
Tel.: 0473(28)6005  
Telex: 2993122 MICRO J.

Ichikawa is just northeast of Tokyo.

These men are both Microware representatives and will serve as distribution agents for information we forward to them. Also, we hope to be kept abreast of the significant happenings and software developments in their areas. Dr. Keil indicated that there are probably 200-300 OS-9 users in Germany, but possibly eight times that in France. An estimate of about 300 users in the U.K. was made. (Please help us to update these estimates.)

In order to keep deliberations short, the choice of Dr. Keil and Mr. Shinjo was quite arbitrary and unilateral. A mechanism to choose such representatives more formally should be a part of next year's meeting agenda.

When software distribution details are pinned down, these contacts will be used to reach their constituencies. In addition, once we have the OS-9 SBS operating on an OS-9 machine, we will try to find an economical way of providing access to it by TYMNET or some similar service which may be useful for our members abroad. We will gladly consider establishing other contacts in countries or continents where the need is felt and where volunteers come forward.

## Membership & Communications

The official mailing address of the OS-9 Users Group is as follows:

OS-9 Users Group  
P. O. Box 8027  
Des Moines, IA 50301

All mail sent to this address will be forwarded to the appropriate officer for action or consideration. Thus, a lag in response may be expected. It was decided, however, that a permanent address was necessary.

If you know another address of the person you wish to reach, you may certainly communicate directly. Please do not publish other addresses for the Group, however.

Membership in the Group may be obtained by sending a check payable to the OS-9 Users Group for \$25 to the above address.

### CompuServe Special Interest Group Jim Bellomo

I have been in communication with the folks at CompuServe and they are enthusiastic but slow. I received descriptive information on OS-9 from Andy Ball at MicroWare and sent it out to CompuServe. They were impressed. At first they thought that Andy had developed OS-9. GADS !!!!!!! Well, they know the history now.

Special Interest Groups on CompuServe function much as do bulletin boards which you may have used. The SIG can be set up to allow Users Group members only to download software. But I have some thoughts on this. The SIG will not cost any money to the OS-9 group. It will only cost the user the connect time (\$5.00 an hour from 6 PM to 5 AM and all day on weekends and holidays their local time.) The CPM SIG works this way: When one first checks on it asks if you would like to become a member. If you answer yes they will process your membership in about a week (there is no cost). Once you are established as a member you are allowed to download programs. If you are not a member you are allowed to view the messages and such things but not download files. There is good security. The user signs on CompuServe with an ID and a password. This information is logged to the SIG the user is on so there should be few problems with security. The people from CompuServe said it will take 6 to 8 weeks to get the SIG underway. That time has now passed, and we may see action this week (the first week of November).

If anyone feels that they have quite a bit of software to upload to the SIG and does not want to bear the cost then they can contact myself or an official of the group and we shall assist. This by no means is to replace the OS-9 users group or software exchange; it is meant to be an enhancement and to educate the nation to OS-9. I would appreciate any comments. I can be reached on the OS-9 BBS, CompuServe (ID 71625,240) or at home at 203-621-5925 8070).

P.S. The above is edited from Jim's message on the OS-9 BBS. When the OS-9 SIG has been put up, it will be noted in the CompuServe sign-on banner ... soon we hope. It will also be noted on the OS-NINE BBS and on the COCO CompuServe SIG.

Subscription to CompuServe cost only \$20 from Radio Shack. You get a free hour of access and a similar opportunity for the Dow-Jones News Service for the money.

## CompuServe Numbers for Officers

Look for the officers on CompuServe using these:

Dale Puckett, Pres.	71446,736
Peter Dibble, V.P.	71426,423
George Dorner Tress	70536,106

We're just beginners at this, but activity should pick up soon.

You can also contact Microware via CompuServe EMAIL. Their number is 73105,1265. MW will try to read mail and respond at least once a week. They will try to answer general inquiries and questions but can't handle detailed technical support (use the Hotline for this IF you are paid up) nor will communications of a commercial nature be carried on here.

### Treasurer's Report

An account has been established for the OS-9 Users Group at St. Paul Federal Savings and Loan Association of Chicago. Funds from last year's treasurer, Terry Strachley, were transferred to this account after deposits from the Microware meeting in Des Moines. Here's a brief report:

	Debit	Deposit
Opening deposit (4 memberships)		\$ 100.00
Transfer T. Strachley		2173.92
Two memberships		50.00
Microware - P. O. Box	\$ 15.00	
Interest		<u>2.61</u>
Balance as of 11/1/83		\$2311.53

Submitted by George Dorner, Treasurer

### Software Exchange

ATTENTION: Much of the software donated in Des Moines was not accompanied by the required paperwork. Signed forms (especially the "Public Domain" statement) are required before the software can be distributed.

#### OS-9 User Group Software Library as of 10/29/83

Title: ACIA.MAPIN By: DIBBLE, PETER  
Language: 6809 Format: SOURCE  
Brief Description: FILTER TO CRANGE CONTROL STRINGS FROM A TERMINAL INTO CORRESPONDING ANSI STRINGS.

Title: ACIA.MAPOUT By: DIBBLE, PETER  
Language: 6809 Format: SOURCE  
Brief Description: APPENDAGE FOR THE ACIA DRIVER; GETS CONTROL OF INIT, READ, AND WRITE AND PASSES THEM THRO ACIA TO THE PHYSICAL DEVICE.

Title: ANSI.GOTOKY By: DIBBLE, PETER  
Language: 6809 Format: SOURCE  
Brief Description: VIDEO TERMINAL DATA MODULE FOR ANSI-STANDARD TERM; FOR USE WITH "DYNASTAR" SCREEN EDITOR.

Title: ATTR\_CHG By: CULP, JERRY  
Language: BASIC09 Format: SOURCE  
Brief Description: "GROUP-CHANGE" OF ATTRIBUTES OF ALL FILES IN A SPECIFIED DIRECTORY.

Title: BINCOM By: KREIDER, C. R.  
Language: BASIC09 Format: SOURCE  
Brief Description: COMPARES TWO SPECIFIED FILES

Title: BLANKO By: CAPOUCH, BRIAN  
Language: BASIC09 Format: SOURCE  
Brief Description: BLANKS A COMPLEX DATA STRUCTURE TO NULLS

Title: BOOTSPLIT By: KREIDER, C. R.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: SPLITS MERGED OBJECT FILES (SUCH AS "OS9800T") INTO SEPARATE MODULES.

Title: CHAR\_TO\_INT\_TO\_CHAR By: CAPOUCH, BRIAN  
 Language: BASIC09 Format: SOURCE  
 Brief Description: TWO PROCEDURES TO CONVERT BASIC09 STRINGS TO INTEGERS AND VICE-VERSA.

Title: CHECK\_FILE By: CAPOUCH, BRIAN  
 Language: BASIC09 Format: SOURCE  
 Brief Description: DETERMINES FILE STATUS OF SPECIFIED INPUT FILE.

Title: DATE\_CVT By: CAPOUCH, BRIAN  
 Language: BASIC09 Format: SOURCE  
 Brief Description: CONVERTS DATE BETWEEN TWO FORMATS: MM/DD/YY TO/FROM YYDDD (WHERE "DOD" IS THE JULIAN DAY OF YEAR FORM 1-365).

Title: DCOPIY By: CAPOUCH, BRIAN  
 Language: BASIC09 Format: SOURCE  
 Brief Description: FULL DIRECTORY COPY UTILITY

Title: DDIR By: SEATON, W. G.  
 Language: 6809 Format: SOURCE  
 Brief Description: LISTS ACTIVE SYSTEM DEVICES, THEIR PATH DESCRIPTOR ADDRESS, PHYSICAL ADDRESS, SYSTEM BUFFER, DEVICE DRIVER & FILE MANAGER

Title: DLIST By: KREIDER, C. R.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: DISK DUMP UTILITY. WORKS LIKE "DUMP /D08", EXCEPT YOU CAN START AT ANY SECTOR ON THE DISK.

Title: DOLLAR\_PRINT By: CAPOUCH, BRIAN  
 Language: BASIC09 Format: SOURCE  
 Brief Description: CONVERTS "REAL" AMOUNT (greater or equal to 999.99 ) TO "STRING" "DOLLAR" FORMAT.

Title: EQDFIX By: KREIDER, C. R.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: STRIPS COMMENTS, BLANK LINES, AND PSEUDO OPS FROM EQUATE FILES

Title: ERREPORT By: CAPOUCH, BRIAN  
 Language: BASIC09 Format: SOURCE  
 Brief Description: PRINTS MESSAGE ON LINE 2 OF TERMINAL, WAITS FOR CMTL-F ACKNOWLEDGEMENT.

Title: FLEXBIN By: KALEITA, DAVID L.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: CONVERTS "FLEX-FORMAT" (COMPRESSED) BINARY FILES INTO FULL IMAGE (FULL SIZE, POSITION-DEPENDENT) BINARY FILES, READY TO "ROM".

Title: FORMS2.GNX By: MORSE, G.  
 Language: CIS COBOL Format: SOURCE  
 Brief Description: STRUCTURED AND MAINTAINABLE 1002 REPLACEMENT FOR MICRO FOCDS FORMS2 PACKAGE. INCLUDES FILES FORMS2.GN1 & F0BMS2.GN2

Title: GETNUMB By: MORSE, G.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: MORE POWERFUL VERSION OF THE BASIC09 "VAL" FUNCTION.

Title: HCOPIY By: SNYDER, HAL, M.D.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: PROMPTING "COPY" UTILITY

Title: HDEL By: SNYDER, HAL, M.D.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: PROMPTING "DELETE" UTILITY

Title: HDIR By: CADMUS, RAY  
 Language: BASIC09 Format: SOURCE  
 Brief Description: HIERARCHICAL "DIRECTORY" UTILITY

Title: HELF By: SEATON, W. G.  
 Language: 6809 Format: SOURCE  
 Brief Description: PRINTS USER INFORMATION ON THE SPECIFIED FILENAME.

Title: KX By: DIBBLE, PETER  
 Language: 6809 Format: SOURCE  
 Brief Description: CONVERTS STANDARD INPUT DATA TO READABLE HEX DUMP FORMAT OUTPUT.

Title: ILEAV By: KREIDER, C. R.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: PRINTS DISK INTERLEAVE TABLES

Title: INKEY\_HAL By: SNYDER, HAL, M.D.  
 Language: 6809 Format: SOURCE  
 Brief Description: READ INDIVIDUAL KEY DEPRESSIONS

Title: INSTALL By: KREIDER, C. R.  
 Language: 6809 Format: SOURCE  
 Brief Description: USED TO "LINK" A FILE CALLED "OS9800T" ON THE SPECIFIED DEVICE. THIS FILE MAY BE PLACED ON THE DEVICE USING A STANDARD "COPT"

Title: JERRYBENCH By: POUHELLE, JERRY / MORSE, G.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: 10x10 MATRIX MULTIPLY BENCHMARK

Title: KILL13 By: SNYDER, HAL, M.D.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: STRIPS ALL "\$13" (X-OFF) CHARACTERS FROM A FILE.

Title: KIMTRON By: KALEITA, DAVID L.  
 Language: 6809 PROCEDURE Format: SOURCE & TEXT  
 Brief Description: SET OF FILES TO AID IN THE USE OF THE KIMTRON ABM85 TERMINAL WITH OS-9. STARTUP FILE WILL AUTO-PROGRAM FUNCT KEYS. DYNASTAR "COTOXY" MODULE FOR THIS TERMINAL IS ALSO INCLUDED.

Title: LIST By: MICROWARE SYSTEMS  
 Language: 6809 Format: OBJECT  
 Brief Description: LISTS A TEXT FILE AND PRINTS TITLE, PAGE NUMBER, AND DATE (WORKS LIKE THE NEW "PR" UTILITY BY MICROWARE).

Title: LISTN By: KREIDER, C. R.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: TRIVIAL PROGRAM TO LIST A TEXT FILE WITH EACH LINE PREFIXED BY A LINE NUMBER.

Title: LLOAD By: SNYDER, HAL, M.D.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: UPLOADS A TEXT FILE, ONE LINE AT A TIME, TO A FULL-DUPLEX BULLETIN BOARD SYSTEM.

Title: MODBUILD By: SNYDER, HAL, M.D.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: PROMPTING FILE "MERGE" UTILITY

Title: MODEM By: MacLEAN, D.  
 Language: 6809 Format: SOURCE  
 Brief Description: COMMAND TO COPY CHARACTERS TO/FROM SPECIFIED DEVICE.

Title: MODLIST By: SNYDER, HAL, M.D.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: PROMPTING "IDENT" UTILITY

Title: MUSIC By: KALEITA, DAVID L.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: AUDIO RECORDING CATALOGING SYS

Title: NETWORK By: KREIDER, C. R.  
 Language: BASIC09 Format: SOURCE  
 Brief Description: ELECTRONIC CIRCUIT NETWORK DESIGN /ANALYSIS

Title: PRINT By: BIALON, LAURA  
 Language: BASIC09 Format: SOURCE  
 Brief Description: FILE PRINTING UTILITY- ADAPTED FROM K&P'S "SOFTWARE TOOLS"



Title: PRINT\_MW By: MICROWARE SYSTEMS  
Language: 6809 Format: OBJECT  
Brief Description: WILL TAB AN ASSEMBLER LISTING  
WITH PAGE NUMBER, TITLE, AND DATE.

Title: PWD By: DIBBLE, PETER  
Language: 6809 Format: SOURCE  
Brief Description: MORE POWERFUL VERSION OF MW'S  
"PWD" COMMAND

Title: PWD\_NAME By: CAPOUCH, BRIAN  
Language: BASIC09 Format: SOURCE  
Brief Description: DOES A "PWD" AND RETURNS THE  
RESULT (OUTPUT) AS A BASIC09-READABLE STRING  
VARIABLE.

Title: QDIR By: KREIDER, C. R.  
Language: BASIC09 Format: SOURCE  
Brief Description: PROMPTING VERSION OF THE NEW MW  
"D" COMMAND (MINUS THE OPTIONS). OUTPUTS A VERTICAL  
DIRECTORY LISTING.

Title: RENOOK By: SNYDER, HAL, M.D.  
Language: BASIC09 Format: SOURCE  
Brief Description: MOVES A FILE FROM ONE DIRECTORY  
TO ANOTHER (ON THE SAME DEVICE) WITHOUT  
"COPY/DELETE"

Title: REMOTE By: SEATON, W. G.  
Language: 6809 Format: SOURCE  
Brief Description: "LINKS" USER TO A SPECIFIED  
REMOTE TERMINAL PATH FOR COMMUNICATION.

Title: RESRATNC By: KREIDER, C. R.  
Language: BASIC09 Format: SOURCE  
Brief Description: PRINTS ALL 1% RESISTOR  
COMBINATIONS THAT FIT THE SPECIFIED RESISTOR RATIO  
AND ERROR (TOLERANCE) LIMITS.

Title: SIEVE By: MORSE, G.  
Language: BASIC09 Format: SOURCE  
Brief Description: SELF-TIMING BENCHMARK

Title: SQSH By: DIBBLE, PETER  
Language: 6809 Format: SOURCE  
Brief Description: "SQUASHES" TEXT FILES BY  
REPLACING EVERY CARRIAGE RETURN CHARACTER WITH A  
"/" AND EVERY STRING OF MORE THAN 1 SPACE WITH 1  
SPACE.

Title: STRIPNUM By: KALEITA, DAVID L.  
Language: BASIC09 Format: SOURCE  
Brief Description: STRIPS A SPECIFIED NUMBER OF  
CHARACTERS FROM THE BEGINNING OF EACH LINE IN A  
TEXT FILE; A NEW FILE IS CREATED AS THE OUTPUT.

Title: STRIPREM By: KALEITA, DAVID L.  
Language: BASIC09 Format: SOURCE  
Brief Description: STRIPS ALL "REM" STATEMENTS OUT  
OF THE SPECIFIED INPUT FILE (WHICH IS NOT  
MODIFIED); A NEW "PROCEDURE" IS CREATED AS OUTPUT.

Title: SYSCALL\_HAL By: SNYDER, HAL, M.D.  
Language: 6809 Format: SOURCE  
Brief Description: EXECUTE AN OS-9 SYSTEM CALL FROM  
BASIC09

Title: TERM\_CNTL By: CAPOUCH, BRIAN  
Language: BASIC09 Format: SOURCE  
Brief Description: PERFORMS SPECIAL TERMINAL  
CONTROL SEQUENCES.

Title: TEXCOM By: KREIDER, C. R.  
Language: BASIC09 Format: SOURCE  
Brief Description: COMPARE TWO TEXT FILES A LINE AT  
A TIME.

Title: TRANSLIT By: BIALON, LAURA  
Language: BASIC09 Format: SOURCE  
Brief Description: (ADAPTED FROM K&P "SOFTWARE  
TOOLS")

Title: WORDS By: DIBBLE, PETER  
Language: 6809 Format: SOURCE  
Brief Description: PARSES INPUT INTO WORDS AND  
OUTPUTS EACH WORD ON A SEPARATE LINE.

#### Late Notice

The CompuServe SIG went public December 1. It  
is on Page PCS-18. A guide to using the OS-9  
Special Interest Group (SIG) software on CompuServe  
has been prepared and is available at no charge  
from the OS-9 Users Group. To receive a copy of  
the guide, send a self-addressed, stamped envelope  
with sufficient postage to cover return mailing  
costs (two units of first class postage) to:

OS-9 Users Group  
P. O. Box 8027  
Des Moines, IA 50301

-----  
Application for membership OS-9 Users Group  
Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
Phone Number \_\_\_\_\_  
We sometime make our membership list available  
to groups we believe our members might be in-  
terested in. Please check here if you do not  
wish to be included in the list we distribute:  
( )  
-----  
Check Type of OS-9:  
Level One( ) Level Two( ) 68000( )  
Hardware manufacturer \_\_\_\_\_  
Floppy Disk size 5 1/4"( ) 8"( ) Other( )  
-----  
\*\*\*\* Send this application with a check or  
or money order for \$25 to:  
-----  
OS-9 Users Group  
P. O. Box 8027  
Des Moines, IA 50301  
-----

## 68000 SAGE SYSTEM

Dear Don,

This letter has been in the works for a long time, but Ron Anderson's  
column in the February issue finally got me off my duff and set me to work.

Like Ron, I too was very dismayed at the 68000 offerings available on  
the S-50 bus. Then I started extending the horizon a little. I had been  
thinking about supplementing my faithful, venerated, but ancient 1975 SK7P  
mainframe with something a little more reliable. After looking over the  
ads in 68 Micro and thinking about was really useful to me, I started  
looking in BYTE. Suddenly people like Jerry Pournelle were talking about a  
new 68000 machine called the SAGE, and it looked good. A little more  
investigation at the SAGE Computer Technology branch office in Boston  
proved that the machine was real and did what it claimed to do. I was  
impressed enough to buy one, and after 9 months I have no regrets at all.

The SAGE II is barely larger than the two half-height floppies on its  
front panel, but size is a deceptive measure of computer power. Inside the  
box is an 8 Mhz 68000 that runs flat-out with no wait states ever. It runs  
BYTE's Sept. 81 benchmark in 4.2 seconds in native code Pascal and 1.12  
seconds in raw machine code. The single system board has sockets for 512K  
of DRAM, two serial ports, a parallel printer port, an IEEE-488 port and a  
disk controller that runs two 640K byte octo-density 5-1/4 inch floppies.  
The NEC floppy controller allows me to configure either of the drives in a  
different disk format to read IBM or Flex disks. If you have more than  
128K of RAM, you get a built in RAMdisk as well.

The system comes bundled with a real Pascal and with an extremely fast UC50 p-system operating system. When I bought the machine it also included the p-system BASIC and FORTRAN. (The new releases have substituted a text editor and spreadsheet.) A single additional disk supplied me with macro cross-assemblers for the 6800, 6803, 6502, 8080, 8085, Z-8, Z-80, 8086, 9900 and LSI-11. In the p-system, when the assembler or compiler detects an error it pops you right into the editor with the cursor pointing to the place in the source code where it got upset. A few seconds and you've fixed the error and are re-assembling. I'd seen this take several minutes on the Apple p-system, but on the SAGE it happens in seconds, even with 30 page source programs. I was able to teach myself Pascal with the same interactivity that BASIC provides. Now I'm experimenting with Wirth's new Modula II language which has all the good parts of ADA without the faults.

What does all this cost, you may ask? The surprising answer is that the SAGE II with 512K of RAM and with the UC50 Pascal, P-system, text editor and spreadsheet costs quite a bit less than just the hardware of a dual disk GIMIX 6809 machine with 12K - less the software. My total investment with Modula, CPM/68K and the cross-assemblers was under \$4500. And does the SAGE have any other software? Well, SAGE recently put out a half inch thick book that lists over 70 major offerings including FORTH, APL, C, the IDRIIS and CPM/68K operating systems and lots of business software. Plus there is a Pascal users library with a lot of public domain stuff. Unfortunately, the offering does not yet include Uniflex. When I talked to TSC, they were hoping that SAGE would hire them to write it. They might be a lot better off doing it on their own. SAGE has plenty of software vendors, but TSC has only done Uniflex 68K for the ExorMACs at this time.

And what about hardware expansion? For the die-hards who have to get at the bus and are willing to leave the lid off the box, there is an internal bus (two 50 pin ribbon connectors) that allows SAGE to add their expansion board with four more UARTs, a hard disk controller, and another 512K of RAM. The hobbyist can use this system bus for hardware hacking. Complete system schematics, ROM listings and system documentation (4 fat books worth) are supplied with the machine.

Do I miss the S-50 bus? Perhaps you might ask if I miss the dirty Molex connectors and the fried power supply diodes from mis-locating boards in the wee morning hours. Actually, I can run the hardware in my SWTP by sending data through the serial port from the SAGE, and I still use the SWTP for Stylograph. It may seem like heresy, but I don't feel that the S-50 bus is adequate or appropriate for the 68000. An 8-bit 68008 implementation seems like a waste of a good processor. I think the S-50 bus was a fine innovation in the year 1975 and is an adequate 8-bit bus for the 80's, but this fixation on a single bus by 68 Micro Journal seems a bit illogical. Come on guys and open up the eyeballs. Are we to miss hearing about SAGE and McIntosh because they aren't S-50. Please, Don, if you're going to put a 68000 on the cover of your magazine, try to include some information about real 68000 systems on the inside.

Sincerely yours,

*Richard H. Lord*  
Richard H. Lord  
Bennett Rd.  
Durham, NH 03824  
(603) 659-2721

Richard, thank you for the letter concerning your SAGE 68000 computer. We are looking for 68K material! However, I would appreciate it if you could go a little more into detail on the SAGE. You know, like most of the articles that appear in 68 Micro Journal.. Not necessarily a comparison article, apples and oranges you know, but what the system does and how well (or how NOT so well). Also ease of use, price and quality of software, documentation, availability of hardware and software and all the other info that most newcomers to the 68000 would want to know.

Your letter covered enough subjects to fill many a page that our readers would surely want to read. Also just some "GENERAL" information on the 68000 and the 68008 would be appreciated. We all (staff here) find that there is not enough input on the 68000 from our readers to justify a full 68000 effort, at this time. However, with input from satisfied users like yourself we could fill many pages with real useful 68K material and thousands of readers would thank you and all that contribute.

Also we expect to see many more 68000 CPU cards and other accessories advertised in 68 Micro Journal in the future. I suspect that shortly we will see 68000 systems from SWTPC, GIMIX, SSB and others as well as HELIX and LSI who are delivering 68000 CPU cards for existing systems now. Also we hear things from MICROWARE (68000 OS-9) and TSC (68000 UNIFLEX) but having seen nothing can't comment.

As to "Operating Systems" it seems that UNIX and CPM 68K are the ones who will dominate the 68000 market unless some of the other OS writers get on the stick! Reputation (?) is one thing delivery is

another, what is needed in the 68000 community right now is delivery of good software that does not expect the end-user to debug it for its creator. We have had too much of that in the past. Fact is that we have had more than one prospective software buyer call us asking about "horror" stories heard about some 6800/6809 software and operating systems. Nowadays it is going to require hardware and software ready to run, and RUN PROPERLY. Bugs and stupid omissions will not be tolerated by todays users. The past is gone and to make it this time around, it better work right the FIRST time!!!!!! Software version number 36 or some such high number, or hardware with wire and piggyback IC sockets won't cut it. Believe me those days are past, forever! Thanks again Richard, please let me know if you feel that you could contribute something suggested above. We are not "rock bound" to the Standard S50 Bus, and we certainly need to keep abreast of all other systems that our readers are interested in. Apparently many if not most have some interest in one or the other variants of the 68000, and we will certainly cover the 68000 in more depth as time passes. It seems that the popularity of a system is only as high as is the availability of "APPLICATIONS" software for said system. The 68000 community has not shown me much today. Looking forward to your reply and trusting you will find time to share your 68000 experiences with the thousands of our readers.

DMW

Dipl. Ing. Jürgen Knauft

SOFTWARE HARDWARE DIGITALELECTRONIC



## COLOR GRAPHICS DISPLAY CONTROLLER



Universal, multimode color and black & white video controller with 512 x 512 dots high-resolution, EF 9365 LSI-graphics display processor chip (1024 x 512 dot-version will be released later)

### BASIC FEATURES version 1 and 2

- 512 x 512 x 3 dots color-coded or 512 x 256 dots monochrome on special order - display
- 180° color-rotated screen memory for 2 different pictures. The three memory planes are individually addressable or in any combination, rotated and stored in one etc.
- EF 9365-LSI graphics display processor high speed vector scan with hatched to animation (up to 1,500,000 dots/s, all planes can be written simultaneously)
- 2 addressable modes of operation: color or monochrome
  - color display with 8 (or 4096) x 1 colors
  - 8 x 8 display with 8 level luminance
  - 8 x 8 addressable modes of 3 memory planes on
- Full ASCII character generator (96 upper and lower case characters, programmable scaling factor in the range of 1:16 times the basic format of 8 x 5 pixels (w/character zoom, independent of programmable horizontal and vertical zoom factors, normal and horizontal or vertical fixed characters, special symbols, max. alphanumeric screen area is 85 x 87)
- Vector color & monochrome addressable types of lines, dots, curves, curves, curves, curves
- Full support of light pen - addressable
- Powerful pointer-like commands for easy programming selected modes and features are valid until they are changed, pen-down or up by command, erase-select makes it possible to erase any dot. Asynchronous access to all registers at all times without waiting for retrace operations for easy software handling and for simple software interface to high level languages
- 16 memory memory operations are done during retrace cycles - so you have always a faster free picture
- Interrupt mode allows external event programming in RQ or FREQ (jumpers selectable on following occurrences: Command ready, lightpen sequence ended, vertical retrace, lightpen switch active)
- 1-4 screen memory addressing for easy crop setting, auto coordinate address advance - address registers points to end of vector (or start of next character after writing a character)





```

1740 PRINT NO, TAB(22);"AMOUNT SPENT ON "IDB#
1750 PRINT NO
1760 PRINT NO, "ITEM":
1770 IF FL=0 THEN GOTO 1830
1780 PRINT NO, TAB(20);"ALLOTTED":
1790 PRINT NO, TAB(36);"SPENT":
1800 PRINT NO, TAB(47);"REMAINING":
1810 PRINT NO, TAB(60);"PERCENT"
1820 GOTO 1840
1830 PRINT NO, TAB(36);"SPENT"
1840 PRINT NO

1850 IF FL=0 THEN GOTO 1960
1860 FOR I=1 TO N1
1870 PRINT NO, S(X);
1880 PRINT NO, USING "#####.##", TAB(20);S(X);
1890 PRINT NO, USING "#####.##", TAB(34);V(X);
1900 IF S(X)-V(X)/O THEN GOTO 1930
1910 PRINT NO, USING "#####.##", TAB(47);S(X)-V(X);
1920 GOTO 1940
1930 PRINT NO, USING "#####.##", TAB(47);S(X)-V(X);
1940 IF S(X)=0 THEN PRINT NO, TAB(63);"?" : GOTO 1760
1950 PRINT NO, USING "#####", TAB(60);V(X)/O/S(X);
1960 NEXT I
1970 GOTO 2030
1980 FOR I=1 TO N1
1990 PRINT NO, S(X);
2000 PRINT NO, USING "#####.##", TAB(34);T(X);
2010 NEXT I
2020 PRINT NO, "-----"
2030 LET W=0
2040 LET R=0
2050 LET M=0
2060 FOR I=1 TO N1
2070 W=W+T(X);
2080 R=R+S(X);
2090 M=M+V(X);
2100 NEXT I
2110 PRINT NO, "TOTAL":
2120 IF FL=0 THEN GOTO 2160
2130 PRINT NO, USING "#####.##", TAB(20);R;
2140 PRINT NO, USING "#####.##", TAB(34);M;
2150 GOTO 2180
2160 PRINT NO, USING "#####.##", TAB(34);W;
2170 GOTO 2240
2180 IF R=M/O THEN GOTO 2210
2190 PRINT NO, USING "#####.##", TAB(47);R-M;
2200 GOTO 2220
2210 PRINT NO, USING "#####.##", TAB(47);R-M;
2220 IF R=0 THEN PRINT NO, TAB(63);"?" : GOTO 2240
2230 PRINT NO, USING "#####", TAB(60);M/O/W;
2240 PRINT NO
2250 IF F#="Y" THEN CLOSE 0
2260 RETURN
2270 IF ERR=0 THEN RESUME 1450
2280 PRINT " FILE NOT FOUND"
2290 RESUME 1540
2300 CLOSE 1
2310 LET N=7
2320 GOSUB 2420
2330 IF ERR=4 THEN PRINT "DISC FILE "IDB#" NOT FOUND"
2340 RESUME 590
2350 LET N=7
2360 GOSUB 2420
2370 PRINT " -> NAME OF THE FILES WERE FOUND: -"
2380 GOTO 590
2390 GOSUB 2420
2400 END
2410 REM SGRGC TO 120 CLEAR SCREEN AND HOME CURSOR
2420 POKE PO,HEX("1B")
2430 POKE PO,HEX("2B")
2440 FOR I=1 TO 2
2450 POKE PO,HEX("00")
2460 NEXT I
2470 IF UB=0 THEN GOTO 2510
2480 FOR I=1 TO N
2490 PRINT
2500 NEXT I
2510 LET UB=-99
2520 LET N=12
2530 RETURN

```

by Joe Turner

Computer Systems Center  
13461 Olive Blvd.  
Chesterfield, MO 63017  
(314) 576-5020

In working with modern disassemblers, such as our DYNAMITE, it is easy to lose sight of the fact that some of the fancy software being disassembled was probably (you never know for sure) written with the aid of macros. Be on the lookout for repetitive blocks of code, and replace them with macros, and your source file will shrink and at the same time become more readable.

One example of where macros could be put to good use is in disassembling TSC's BASIC and XBASIC interpreters, which use a special code to represent keywords. (Does anybody know why they do that?) Replacing the hundreds of FCB's generated

by the disassembler with macro calls will make the strings much easier to read and (if you're really brave) add to and/or modify.

I have turned on the 'EXP' option to show the macros expanded. Normally you turn off this option after you are sure that the macros work the way you want them to work.

The example given here is admittedly crude. If you know of a more elegant way of doing the same thing with the TSC 6809 Macro Assembler, please let me know.

This is just one more example of the fact that, even with the best disassembler software package, some creative thought is required of the human disassembler.

```

MACRO      .      Call it whatever you like.
IFC        &1,1    IF first parameter is '1',
FCB        ' &2*2-26 then assemble this.
ENDIF

IFC        &1,2    and so on...
FCB        ' &2*2-26
FCB        ' &3*2-26
ENDIF

IFC        &1,3
FCB        ' &2*2-26
FCB        ' &3*2-26
FCB        ' &4*2-26
ENDIF

IFC        &1,4
FCB        ' &2*2-26
FCB        ' &3*2-26
FCB        ' &4*2-26
FCB        ' &5*2-26
ENDIF

IFC        &1,5
FCB        ' &2*2-26
FCB        ' &3*2-26
FCB        ' &4*2-26
FCB        ' &5*2-26
FCB        ' &6*2-26
FCB        ' &6*2-26
ENDIF

IFC        &1,6
FCB        ' &2*2-26
FCB        ' &3*2-26
FCB        ' &4*2-26
FCB        ' &5*2-26
FCB        ' &6*2-26
FCB        ' &7*2-26
ENDIF

IFC        &1,7
FCB        ' &2*2-26
FCB        ' &3*2-26
FCB        ' &4*2-26
FCB        ' &5*2-26
FCB        ' &6*2-26
FCB        ' &7*2-26
FCB        ' &8*2-26
ENDIF

IFC        &1,8 Do some error trapping.
ERR        STRING ENCODE ERROR (ENC 8)
ENDIF

```



\*\*\*\* FACT SHEET \*\*\*\*

**ESM Electronic Specialists, Inc.**  
171 South Main Street, North, Mass. 01760  
(617) 866-1637

By going through a fairly involved procedure, you may save ROM packs off to disk and load and execute them if you know the beginning, execution and ending addresses. The programs have to be relocated, since the ROM packs use memory at 0000 and so does the DOS. Fortunately the romable code from all of the packs I have tried so far is easily relocatable.

The procedure is to use an optional cable from Esatron which sockets into the expansion bus on the Color Computer with the ROM pack you wish to copy in place on the cable. Using tape I/O you backup the ROM onto tape and then unplug the ROM pack cable and plug in the disk drive interface. After booting into the system you load the ROM pack image off of the cassette into memory where you want it to execute and then save it to disk as a binary file. Executing is a simple matter of load and execute.

The functions provided by the on board monitor are as follows:

- "A" - display contents of memory in hex and ASCII.
- "B" - set breakpoint.
- "C" - display memory in hex.
- "D" - go to a memory location and execute.
- "E" - load CCDS from disk into memory.
- "F" - modify memory.
- "G" - save memory block to address in memory.
- "H" - test all of RAM (includes user and DOS RAM - very good test).
- "I" - save a constant into a block of memory. Used for clearing memory, etc.
- "J" - exit and return to BASIC.

While this is not quite the best monitor available by any means, it certainly is the handiest since it is always there.

The only real complaint I have about the system relates to the cable from the interface to the computer. This is a ribbon cable with a male connector on the end. Esatron would do well to consider putting a case on the cable such as found on the ROM packs. As it is, it is a bit difficult for those with ten thumbs (such as I or self) to plug it in. The save complaint holds true for the ROM backup master.

Don't take the above complaint too seriously. All and all, I'm tickled pink with the interface. The documentation is certainly adequate for one who has upgraded from a Mod-1 to a Color Computer, even if it is the temporary manual.

Knowing Esatron from past experience, I would say that their final manual will be great even for the beginner.

Probably the vast majority of Color Computer owners will buy Radio Shack's disk assembler rather than Esatron's, but anyone, like me, who is moving up from the Mod-1 to the Color Computer will want to make sure that at least their BASIC programs will be able to be transferred into the Color Computer. With Esatron you can do this.

Thank you for providing your magazine as a forum for CC owners. Even at this late date in the Color Computers life, there is not much information available for this machine and every bit helps.

Esatron is located at 101 Commercial Street, Sunnyvale, CA. 94086. They have a toll free number for user entry. It is 1-800-338-6399.

John Cline  
The Computer Store, Inc.  
6949 South Peoria Avenue  
Tulsa, Oklahoma 74109

P.S. Sorry about the copy on computer paper, but it was all I had.

- b, ticky - no restraint on printing  
y113 - lje -

## SUPPORT YOUR ADVERTISERS



### SPELL 'N FIX II NOW FREE

NO MONEY RELEASE - February 6, 1984

In releasing their brand new SPELL 'N FIX II proofreader program, Star-Kits Software Systems Corporation announced today that the program will be sent at no charge to anyone who sends Star-Kits a blank disk and a self-addressed stamped disk master. The program as well as all documentation will be on the disk, and users may print their own manual from the disk.

In announcing the plan, Star-Kits' president Peter Stark also indicated that SPELL 'N FIX II users will be encouraged to provide free copies to friends. The program is accompanied by a request that satisfied users make their own estimate of the program's worth to them, and then send a proportionate contribution to Star-Kits to encourage the further development of such programs.

# Keep your PC sane

Garbled data, memory loss, video interference and mechanical malfunctions are among the symptoms of a PC that's "a little touched" by static. A charge in your finger so small that you can't see or feel it can drive a computer crazy. Even when the humidity is up.

Now a **3M FIRST TOUCH™ Static Control Computer Pad** can drain away static the first and every time you touch it.

Keep your PC sane. Pick up a cushioned 3M pad.

## \$59.95

WE PAY SHIPPING



## DATA-COMP

P.O. Box 794 HIXSON, TN 37343  
TOLL FREE 1-800-338-6800

Stars indicated that SPELL 'N Fix II had already been advertised at a price of \$69.29, and was superior to many proofreading programs selling for \$200-300 for other computers. He said that the decision to provide SPELL 'N Fix II free was a difficult one, but that it was a better alternative than strong copy protection. Star-Kits had recently developed an almost break-proof copy protection algorithm, but felt that it would impose too much of a burden on legitimate purchasers. He said, "Based on our experience with the original SPELL 'N Fix, we're convinced that every Code owner will get copies of the new SPELL 'N Fix in a relatively short time. We're sure they will love it, and only hope that enough of these will actually send us a contribution."

More information on SPELL 'N Fix II may be obtained from Star-Kits Software Systems Corporation, P. O. Box 209, Mt. Kisco NY 10549, or by calling (914) 241-0287.

#### STAR-DOS LEVEL I

NEW PRODUCT RELEASE - February 6, 1984

Star-Kits Software Systems Corp. today announced the release of STAR-DOS Level I, a single-user disk operating system for 6809-based computer systems.

The Level I release is a successor to the STAR-DOS which has been available for over a year on the Radio Shack Color Computer. But the Level I version has been rewritten for use on the larger 6809 systems which are popular in industrial, control, and advanced hobbyist applications. This includes SS-50 systems and single-board computers from a variety of vendors.

The modularity of STAR-DOS allows easy interfacing to a variety of hardware, including up to ten floppy or Winchester drives. It has been specially configured to be compatible with current software almost all 6809 programs written for linked-chain disk operating systems (such as Technical Systems Consultants' Flex) will run under STAR-DOS without modification. This includes a variety of language compilers and interpreters, editors and assemblers, word processing programs, financial programs, and debugging packages. Programs which work with STAR-DOS are available from STAR-Kits and dozens of other sources.

STAR-DOS Level I is available both for individual purchase and for licensing to equipment manufacturers. The end-user price in single quantities is \$75, which includes an extensive programmer's manual with sample programs and utilities.

More information on STAR-DOS Level I may be obtained from Star-Kits Software Systems Corporation, P. O. Box 209, Mt. Kisco NY 10549, or by calling (914) 241-0287.

#### COMPILER EVALUATION SERVICES By: Ron Anderson

The S.E. MEDIA Division of Computer  
Publishing Inc.  
Is offering the following **SUBSCRIBER  
SERVICE:**

#### COMPILER COMPARISON AND EVALUATION REPORT

Due to the constant and rapid updating and enhancement of numerous compilers, and the different utility, appeal, speed, level of communication, memory usage, etc., of different compilers, the following services are now being offered with periodic updates.

This service, with updates, will allow you who are wary or confused by the various claims of compiler vendors, an opportunity to review comparisons, comments, benchmarks, etc., concerning the many different compilers on the market, for the 6809 microcomputer. Thus the savings could far offset the small cost of this service.

Many have purchased compilers and then discovered that the particular compiler purchased either is not the most efficient for their purposes or does not contain features necessary for their application. Thus the added expense of purchasing additional compiler(s) or not being able to fully utilize the advantages of high level language compilers becomes too expensive.

The following COMPILERS are reviewed initially, more will be reviewed, compared and benchmarked as they become available to the author:

PASCAL "C" GSPL WHIMISCAL PL/9

Initial Subscription - \$39.95  
(Includes 1 year updates)  
Updates for 1 year - \$14.50

S.E. MEDIA - CPI  
5900 Cassandra Smith, POB 794  
Hixson, TN 37343  
615 842-4601

Dear Don,

Thanks for the colorful letter reminding me that I needed to write this note. Please find enclosed my check for another year's subscription to 68 Micro.

I was reading 68 Micro for about 6 months before I got my Color Computer, and now with DS9 and real languages running on it, I find 68 Micro to be the only true source of solid technical information for DS9 and the 6809 in general. I look forward to staying in the fold.

Thanks for the good job that you're doing and keep up the good work!

Mike Michnovicz  
504 B Ortiz S.E.  
Albuquerque, NM  
87108

#### CLASSIFIED ADVERTISING

\*\*\*

LIKE NEW - ADOS-VIEWPOINT CRT TERMINALS - DEMO UNITS, THESE ARE LIKE NEW - WARRANTY NEVER REGISTERED - SAVE - ONLY \$478.00 EACH - ONLY 1 LEFT. CALL ASK FOR DON, TOM or LARRY

A/C 800 338-6800

\*\*\*

TELETYPE Model 43 PRINTER - with serial (RS232) interface, and full ASCII keyboard. LIKE NEW - New cost \$1295.00 - ONLY \$759.00 ready to run - Call Tom - Larry - Bob, CPI 615 842-4600

\*\*\*

"Complete SWTP 6809 system 2-5 1/4" disks, 56K RAM, 80 x 24 video, keyboard, 12" monitor, NEC 8023A printer, MP-R, clock-timer, software, manuals. \$1495.00. Curt Barrett (301) 577-2105

\*\*\*

GIMIX #28 5 1/4 Disk Controller, Never Used with GIMIX FLEX \$150; SWTPC 8K Memory Board \$35; SWTPC 32K Universal Memory Board (unpopulated) Never Used \$50; Call 805-968-7688 (after 6 PM) CRAIG or Leave Tape Message; 221 Hillview Drive, Goleta, CA 93117; Reasonable offers considered.

\*\*\*

64K Radio-Shack Color C puter, upper/lower case, Disk controller, 2 disk drives w/cable. DATA-COMP version FLEX9, FMATE(RS), TPS6809, TABULA RASA, XBASIC, others. \$1,000. Printer available. Garrett 214-722-0043.

\*\*\*

## STAR-DOS LEVEL I

Whenever a new DOS is introduced, there's always the problem of developing software to work with it. So we did it the opposite way — we analyzed the requirements of software that already exists and developed a DOS that met them... and exceeded them! The result is STAR-DOS Level I, a new DOS for 6809 systems, ideal for single-user industrial, control, and advanced hobbyist applications. This includes SS-50 systems and single-board computers from a variety of vendors.

Level I is compatible with most current 6809 hardware and software. On the hardware side, it allows up to ten floppy or Winchester drives with appropriate controllers. On the software side, it runs existing 6809 software from all the major 6809 software suppliers, including TSC, Star-Kits, Introl, and others.

Write or call for more information. STAR-KITS Software Systems Corporation, P.O. Box 209, Mt. Kisco N.Y. 10549 (914) 241-0287.



# GOOD NEWS!



## **C for the 6809 WAS NEVER BETTER!**

### **INTROL-C/6809, Version 1.5**

Introl's highly acclaimed 6809 C compilers and cross-compilers are now more powerful than ever!

We've incorporated a totally new 6809 Relocating Assembler, Linker and Loader. Initializer support has been added, leaving only bitfield-type structure members and doubles lacking from a 100% full K&R implementation. The Runtime Library has been expanded and the Library Manager is even more versatile and convenient to use. Best of all, compiled code is just as compact and fast-executing as ever - and even a bit more so! A compatible macro assembler, as well as source for the full Runtime Library, are available as extra-cost options.

Resident Introl-C/6809 compilers running under Flex or OS9 are priced from \$375; Uniflex, from \$425.

Cross-compilers for PDP-11/Unix hosts are priced from \$1500.

Trademarks:

Introl-C, Introl Corporation:

Flex and Uniflex, Technical Systems Consultants;

OS9, Microware Systems;

PDP-11, Digital Equipment Corp.;

Unix, Bell Labs.

For further information, please call or write.

**INTROL**  
CORPORATION

647 W. Virginia St.  
Milwaukee, WI 53204  
(414) 276-2937

# OS/9, FLEX, COCO, UNIFLEX SOFTWARE

**SUPER SLEUTH DISASSEMBLER** EACH \$99-FLEX, \$101-OS/9, \$100-UNIFLEX  
interactively generates source on disk with labels, includes xref, label definition, binary file editing, etc.  
specify 680x/6502 version or Z-80/8080/85 version

**(OBJECT ONLY)** EACH \$50-FLEX & OS/9, \$49-COCO DOS

COCO DOS available in 680x/6502 version only

**CROSS-ASSEMBLERS** EACH \$50-FLEX, \$55-OS/9, \$60-UNIFLEX, ALL \$100

specify for 6800/1, 6502, 6805, Z-80, or 8080/48/85

OS/9 version requires Microware RMA or FHL OSM macro assembler

FLEX version requires TSC ASMB or FHL ASM or OSM macro assembler

**DEBUGGING SIMULATORS** EACH \$75-FLEX, \$100-OS/9, \$80-UNIFLEX

specify for 6800/1, 6805/148805, or 6502

**OS/9 AND UNIFLEX SIMULATORS** EACH \$100-FLEX

debug OS/9 and UNIFLEX application programs under FLEX using TSC DEBUG facility

**6502 TO 6809 ASSEMBLER TRANSLATOR** \$75-FLEX, \$85-OS/9, \$80-UNIFLEX

translates 6502 programs to 6809, noting inexact conversions

**6800 TO 6809 & 6809 PIC TRANSLATORS** \$50-FLEX, \$75-OS/9, \$60-UNIFLEX

translates 6800 programs to 6809, 6809 programs to PIC

**FULL-SCREEN FLEX AND UNIFLEX TSC X BASIC PROGRAMS FOR 6809**

(with complete cursor control)

DISPLAY GENERATOR/DOCUMENTOR \$50-FLEX, \$75-UNIFLEX

MAILING LIST SYSTEM \$100-FLEX, \$110-UNIFLEX

INVENTORY WITH MRP \$100-FLEX, \$110-UNIFLEX

TABULA RASA SPREADSHEET \$100-FLEX, \$120-UNIFLEX

**DISK UTILITY PROGRAM LIBRARY** \$50-FLEX

edit disk sectors, sort directory, maintain master catalog, etc.

**CMODEM TELECOMMUNICATIONS PROGRAM** \$50-FLEX & OS/9 & UNIFLEX

provides menu-driven telecommunications facilities, with terminal mode, up/down load, MODEM7 protocol, etc.

**5.25" SOFT-SECTORED DISKETTES** EACH SET OF 50 \$75

with Tyvek jackets, hub rings, labels, write-protect tabs, CSC guarantees DSDO

Specify operating system, computer make and type, terminal type.  
Programs provided in source form on diskette, specify size and density.  
Contact CSC for full catalog and dealer info, printed manuals provided with products.  
For VISA and MASTER CARD, give account, exp. date, phone. US funds only.  
Add 5% for shipping software, but not for diskettes.  
(UNIFLEX trademark) Technical Systems Consultants OS/9 trademark Microware

**Computer Systems Consultants, Inc.**  
1454 Latta Lane, Conyers, GA 30207  
Telephone Number 404-483-1717/4570

## SOFTWARE FOR THE HARDCORE

\*\* FORTH PROGRAMMING TOOLS from the 68XX&X \*\*  
\*\* FORTH specialists — get the best!! \*\*

NOW AVAILABLE — A variety of rom and disk FORTH systems to run on and/or do TARGET COMPILATION for

6800, 6301/6801, 6809, 68000, 8080, Z80

Write or call for information on a special system to fit your requirement.

Standard systems available for these hardware—

EPSON HX-20 rom system and target compiler  
6809 rom systems for SS-50, EXORCISER, STD, ETC.  
COLOR COMPUTER  
6800/6809 FLEX or EXORCISER disk systems,  
68000 rom based systems  
68000 CP/M-68K disk systems, MODEL II/12/16

iFORTH is a refined version of FORTH Interest Group standard FORTH, faster than FIG-FORTH. FORTH is both a compiler and an interpreter. It executes orders of magnitudes faster than interpretive BASIC. MORE IMPORTANT, CODE DEVELOPMENT AND TESTING is much, much faster than compiled languages such as PASCAL and C. If Software DEVELOPMENT COSTS are an important concern for you, you need FORTH!

firmFORTH™ is for the programmer who needs to squeeze the most into roms. It is a professional programmer's tool for compact rommable code for controller applications.

► iFORTH and firmFORTH are trademarks of Talbot Microsystems  
► FLEX is a trademark of Technical Systems Consultants, Inc.  
► CP/M-68K is trademark of Digital Research, Inc.

iFORTH®  
from TALBOT MICROSYSTEMS  
NEW SYSTEMS FOR  
6301/6801, 6809, and 68000

---> iFORTH SYSTEMS <---

For all FLEX systems: GIMIX, SWTP, SSB, or EXORCISER Specify 5 or 8 inch diskette, hardware type, and 6800 or 6809.

\*\* iFORTH — extended fig FORTH (1 disk) \$100 (\$15)

with fig line editor,

\*\* iFORTH + — more! (3 5" or 2 8" disks) \$250 (\$25)

adds screen editor, assembler, extended data types, utilities, games, and debugging aids.

\*\* TRS-80 COLORFORTH — available from The Micro Works

\*\* firm FORTH — 6809 only. \$350 (\$10)

For target compilations to rommable code.

Automatically deletes unused code. Includes HOST system source and target nucleus source. No royalty on targets. Requires but does not include iFORTH +.

\*\* FORTH PROGRAMMING AIDS — elaborate decompiler \$150

\*\* iFORTH for HX-20, in 16K roms for expansion unit or replace BASIC \$170

\*\* iFORTH 68K for CP/M-68K 8" disk system \$290  
Makes Model 16 a super software development system.

\*\* Nautilus Systems Cross Compiler

— Requires: iFORTH + HOST + at least one TARGET:

— HOST system code (6809 or 68000) \$200

— TARGET source code: 6800-\$200, 6301/6801—\$200

same plus HX-20 extensions— \$300

6809—\$300, 8080/Z80—\$200, 68000—\$350

Manuals available separately — price in ( ).  
Add \$6/system for shipping, \$15 for foreign air.

TALBOT MICROSYSTEMS 1927 Curtis Ave., Redondo Beach, CA 90278 (213) 376 9941

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

**SOUTH EAST  
MANIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
for information  
call (615) 642-4601

CoCo OS-9™ FLEX™  
**SOFTWARE**

## ASSEMBLERS

TSC

Macro Assembler  
the FLEX STANDARD Assembler

F,CCF \$50.00

Relocating Assembler w/Linking Loader  
Use with many of the C and Pascal Compilers.

F,CCF \$150.00

Great Plains Comp. Co.

IBMPC

Relocating, Relocating-Macro Assembler  
and Linking Loader for the 6809

Supports Recursive Macros, Conditional Assembly, etc. Optional  
X-Ref Listing. Includes a Small Line-oriented Editor as part of  
the Assembler.

F,CCF \$150.00

F,CCF w/Source \$200.00

OmegaSoft

PRALL

Relocating Assembler and Linking Loader

2-pass Relocating Assembler; 2-pass Linking Loader. Supports 6  
Char. Labels, System Calls (SWI(x) PCB x), Expressions with  
Arith., Logic, and Shifts, etc.

F,CCF \$125.00

One Year Maint. \$50.00

Mindrush Micro Systems

MACE

By Graham Trott. A combined Editor and Assembler designed to  
allow the Programmer to Enter, Edit, and Assemble Programs with  
a minimum of effort, w/o leaving the Program. Designed  
primarily for small to medium sized Program Development.  
Includes SOURCE, a Cross Assembler for the MC6800/1/3 and  
Hitachi HD6301 (OAS 6801).

F,CCF - \$90.00

## DISASSEMBLERS

Computer Systems Consultants

SUPER SLEUTH

Computer Systems Consultants Super Sleuth is a "Time Tested",  
reliable, PROVEN Disassembler that has gained acceptance  
through out the 88-50 Bus Community as an extremely POWERFUL  
INTERACTIVE, Software Tool. The Super Sleuth Software Package  
consists of 3 Programs; SLEUTH (the Disassembler), CHANGES (used  
to globally Change Labels to a meaningful Name), and XREF (a  
Cross Reference Generator for Source Code Files). SLEUTH will  
Disassemble Memory Resident 6809 Code and 6800, 6801, 6802,  
6803 (the "Baby CoCo"), 6805, 6808, 6809, and 6802 (Apple,  
Atari, Commodore, etc.) Binary Disk Files. (See Aug. '83 '68'  
Micro Journal "Color Users Notes" Column for a full Review.)

Color Computer 68-50 Bus (all w/ Source)

CCD (32K Req'd)

Obj. Only \$49.00

CCF, Obj. Only \$50.00

CCF, w/Source \$99.00

CCD, Obj. Only \$50.00

F, \$99.00

U, \$100.00

O, \$101.00

ALL Computer Systems Consultants Software  
runs on the Color FLEX Systems

ALL in stock  
call 800-338-6800  
for IMMEDIATE DELIVERY

Computer Systems Center

OS-9™ +

An "easy to use", powerful MacroAssembler for Disk Resident 6809  
and 6800 Binary Files. Allows the development of a "Control  
File" of various Program "Boundaries" during successive  
disassemblies; can use a Label File which automatically replaces  
a Hex location with a Label Name; includes an XREF Utility; etc.  
Label Files provided for Mini-FLEX, FLEX2, FLEX9, Color Computer  
(for use with Color FLEX Systems), etc. OS-9 Version includes  
special OS-9 options.

CCF, Obj. Only	\$100.00
CCD, " "	\$150.00
F, " "	\$100.00
O, " "	\$150.00
U, " "	\$200.00

## COMPILERS & DECOMPILERS

6809 "Structured" Assembly Lang. Compilers

Mindrush Micro Systems

PL/9

By Graham Trott. A "Structured" Assembly Language  
Editor/Compiler/Debugger, all in ONE PACKAGE: provides a totally  
INTERACTIVE Program Development Cycle. The Compiler supports  
large Symbol Names, Variable Types, Pointers, Control  
Structures, Stack, A-, B-, and D-Register manipulation, etc.  
The Source-oriented Trace/Debugger provides Single Stepping,  
Breakpointing, etc. An excellent software development Tool for  
utilizing the power of the 6809 in developing small to medium  
sized packages.

F, CCF - \$190.00

Whimsical Developments

WEDSICAL

Need the Ease of Design and Maintainability of "Structured  
Programming" AND the Speed and Control of Assembly Language?  
Then WEDSICAL was designed for you! This Single Pass,  
Recursive Descent Compiler provides the tool for developing  
simple Utilities to MAJOR Systems in Assembly Language.  
Supports 3 "Lex" Levels which allow one level of Procedure  
nesting, or more within "Modules". It is easy to develop  
programs written for other machines since you are working at  
the Assembly Language level. Features unified, user-defined  
I/O; produces relocatable, recursive, re-entrant Code;  
Structured style and statements with Procedures and Modules;  
supports Byte and Double-Byte primitives with 3 types of  
Integers (up to 32 bit), Char and Boolean, and unlimited sized  
Arrays (vectors only); Interrupt handling; unlimited length  
Variable Names; Variable Initialization (defaults to \$00);  
Include "Source File" directive; Conditional compiling; direct  
Code insertion; control of the Stack Pointer; etc. To quote  
Ron Anderson in his review of WEDSICAL in the Sept. '83 issue  
of '68' Micro Journal that, except for the lack of floats, "...  
I have to give this one VERY high rating. ...". It is a FAST  
Compiler which produces FAST Code (his "Virus" Benchmark ran  
at 9 secs. on a 2 Mhz System).

F and CCF - \$195.00

'C' Compilers

Mindrush Micro Systems

C Compiler

By James McCoosh. Full featured C Compiler for the FLEX  
Operating System. Includes a Reloc. Asmb., but needs the TSC  
Relocating Assembler/Linking Loader (which includes a Library  
Manager) for those "full blown" System Packages.

F and CCF - \$295.00

Introl

C Compiler

A full-featured C, streamlined for the 6809. Generates very  
efficient object code. Output "benchmarks" close to 10000  
68000 in 8 Bit Operations; 1.5 times faster than a 4 Mhz 286  
when using a 2Mhz 6809 System (Re. p 43, '68' Micro Journal, May  
'83). Floats, etc.

F, CCF, and O - \$375.00

U - \$425.00

One Year Maint. - \$100.00



\*FLEX is a trademark of Technical Systems Consultants  
\*OS9 is a trademark of Microware

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

**SOUTH EAST  
MANIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
info (615) 842-4601

CoCo OS-9™ FLEX™  
**SOFTWARE**

Availability Legend ---

F = FLEX, CCF = Color Computer FLEX  
O = OS-9, CCD = Color Computer OS-9  
U = UNIFLEX  
CCD = Color Computer Disk  
CCF = Color Computer Tape



## PASCAL Compilers

TBC

### PASCAL Compiler

Native Code Compiler (UCSD Oriented).

F and CCP - \$289.00

Lucidata

### PASCAL Compiler

P-Code Compiler (ISO Standard). Designed especially for Microcomputer Systems: Run-time System checks available resources for each task, allowing operation on even minimal computer systems. Allows linkage to Assembly Code for maximum flexibility.

F and CCP 5" - \$199.00  
F 8" - \$285.00

OmegaSoft

### PASCAL Compiler

For the PROFESSIONAL: ISO Based, Native Code Compiler. Primarily for Real-Time and Process Control applications. Use custom I/O devices in place of the Pascal INPUT and OUTPUT; Long Int. (32 Bit); Dynamic length strings; Interrupt processing, ROM-able, PIC, Re-entrant Code, etc. **POWERFUL** Includes Source for the Symbolic Debugger, Runtime, and several Utilities. Requires a "Motorola Compatible" Relocating Assembler and Linking Loader.

F and CCP - \$425.00  
One Year Maint. - \$188.00

## DISCOMPILERS

Southeast Media

### DUB (A UNIFLEX "basic" De-Compiler)

Re-Create a Source Listing from UNIFLEX Compiled **basic** Programs. Easy to Use; Works w/ ALL Versions of UNIFLEX **basic**; Output to Disk or Terminal. Time TESTED and PROVEN: **SOLID!**

U - \$219.95

## UTILITIES

Southeast Media

### BASIC9X JMR

This **BASIC9X Cross Reference Utility** is a BASIC99 Program which will produce a "pretty printed" listing with each line numbered, followed by a complete cross referenced listing of all variables, external procedures, and line numbers called. Also included is a **Program List Utility** which outputs the listing without the overhead of building the cross reference table, which allows it to run considerably faster when only a "pretty printed" listing with line numbers is desired. Requires BASIC99 or RUNB for operation.

```

72 EDITOR (OBJ,INPATH) THEN FOUND:=FALSE; GOTO 117
73 GET INPATH:PHASE
74 GET INPATH:PHASE := GET INPATH:PHASE
75 NEXT INPATH:INSTR:=PHASE
76 PHASE:=1
77 REPEAT
78   GET INPATH:CHART
79   PHASE:=PHASE+1; GOTO 74 UNTIL (CHART=0) OR (PHASE=127)
80 UNTIL (CHART=127)
81 UNTIL (PHASE=0)
82 RETURN

```

Phase	3	26	68	78	74
Name	3	15	28	81	
OutPath	4	51	54	56	
CHART	4	20	30	53	61 62 64 65 66
PHASE	78	79	80		
PHASE	4	22	60	72	
PHASE	8	11			
PHASE	11	13			

O and CCC = Obj. Only - \$39.95  
O and CCC = w/ Source - \$79.95

Southeast Media

### OS-9 vdisk

Give your OS-9 Level 1 System the speed of memory access that can be several orders of magnitude over your present floppy disk drive. Use that **Extended Memory** capability of your SM/PC or GMAx CPU card (or any other that has the same format DMT). The size of the Virtual Disk is completely variable in whole increments of 4K up to 960K, which is all that these systems can address beyond the base page that OS-9 Level 1 uses. By putting all of your OS/2 Directory on your Virtual Disk you can have the fastest execution speed possible (next to eating up System Memory with all of them). You can also set up high speed inter-process communications via random virtual disk files and not eat up valuable system memory with pipe buffers. **Some Assembly Required - Level 1 ONLY.**

O, obj. only - \$79.95  
w/ Source - \$109.95

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

# SANTA EAST MEDIA

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
for information  
call (615) 842-4601

**CoCo OS-9™ FLEX™**  
**SOFTWARE**

Southeast Media

### O-P

#### OS-9 to FLEX - FLEX to OS-9

Finally: the barrier has been removed between OS-9 and FLEX formatted disks! Now you can **READ from**, and **WRITE to**, a Single Sided 5" or 8" FLEX diskette from OS-9 with O-P. O-P is a new and unique program, written in BASIC99 (with Source), that performs the following functions:

**REFORMAT** A BASIC99 Program that reformat a chosen amount of an OS-9 disk to FLEX Format so it can be used normally by FLEX.

**FLEX** A BASIC99 Program that does the actual read or write function to the special O-P Transfer Disk, all selected from a user-friendly menu. Functions provided include reading the FLEX Directory, Deleting FLEX Files, Copying both directions, etc. All selections are interactive and complete, including all necessary prompts to the operator.

FLEX users can read, write and use the special disk as any other FLEX disk, provided the FLEX directory is not allowed to continue beyond track zero (too many files).

F and CCP - \$79.95

Southeast Media

### COPIER

#### Copy LARGE Disks to several smaller disks

The following FLEX utilities allow the backup of ANY disk to any SMALLER size diskettes (Manchester to 8's or 5's, 8" to 5's, etc.). By simply inserting diskettes as requested by COPIER, a large disk system may be downloaded to your present floppy disk system, any size. No need to fiddle with directory deletions or any of the other tedious operations that must be done using the normal copy routines.

COPIER.CHD understands normal "copy" syntax and always keeps up with files already copied by maintaining directories for both host and receiving disk system, eliminating hours of tedious keyboard entries and other time consuming cleanup chores.

BACKUP.CHD is a special program that downloads "random" type files, any size.

RESTORE.CHD a special program to restructure copied "random" files for copying, or recopying back to the host system.

FRAGMENT.CHD a "bonus" utility that "relinks" the free chain of floppy or hard disk thereby eliminating fragmentation.

**Completely documented source files included.**

ALL 4 Programs (8" or 5") \$99.95

Southeast Media

### CHESS 6809

Requires FLEX and DISPLAYS On Any Type Terminal

Features:

Four levels of play.

Swap sides. Point scoring system.

Two display boards. Change skill level.

Solve Checkmate problems in 1-2-3-4 moves.

Make move and swap sides. Play white or black.

This is one of the **strongest** CHESS programs running on any microcomputer, estimated USCP Rating 1600+ (better than most 'club' players at higher levels).

F and CCP - \$79.95



\*FLEX is a trademark of Technical Systems Consultants  
\*OS-9 is a trademark of Microware

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

# SANTA EAST MEDIA

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
info (615) 842-4601

**CoCo OS-9™ FLEX™**  
**SOFTWARE**

#### Reliability Legends

F = FLEX, CCP = Color Computer FLEX

O = OS-9, CCC = Color Computer OS-9

U = UNIFLEX

CCC = Color Computer Disk

CTT = Color Computer Tape

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 for information  
 call (615) 842-4801

**CoCo OS-9™ FLEX™**  
**SOFTWARE**

#### Southeast Media

##### DIET-TRAC Forecaster

**DIET-TRAC** Forecaster is an X-BASIC program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C P G) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individual are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's weight for a 30-day period is printed.

F - \$59.95  
 U - \$89.95

#### Southeast Media

##### UDCA

A **COMMUNICATION** Package  
 for the **UNIFLEX** Operating System

Allows **UNIFLEX** Based Systems to Transmit and Receive files to and from other Computer Systems via Modem. Use with CP/M, Main Frames, other UNIFLEX Systems, etc.

- Verifies Transmission Integrity using checksum or CRC
- Automatically Re-transmits bad blocks
- Transmits data in 128 byte blocks

U - \$299.99

#### Southeast Media

##### JUST

Text Formatter

**JUST**, a Text Formatter developed by Ron Anderson, provides numerous features which make it a valuable addition to any FLEX Users Software Library. **JUST** is designed for formatting Text Output for Dot Matrix Printers and provides many unique features:

- Output the "Formatted" Text to the Display for format analysis and change.
- Output the "Formatted" Text to a Text File for use with the supplied **PRINT.OBJ** for producing multiple copies of the Text on the Printer **INCLUDING** IMBEDDED PRINTER COMMANDS (this Utility is very useful at other times also, and worth the price of the program by itself).
- "User Configurable" for adapting to other Printers (comes set up for Epson MX-80 with Graftex); provides for up to ten (10) imbedded "Printer Control Commands", such as italics on and off, boldface on and off, etc.
- Automatic compensation for a "Double Width" printed line.
- Includes the normal line width, margin, indent, paragraph, space, vertical skip lines, page length, page numbering, centering, fill, justification, etc.
- Use with **ANY** Editor.
- Supplied with "Structured Source" (Mindrush PL/9); easy to see the flow of the program.

F and CCP - \$49.95

#### Lucidata

##### PAROM UTILITIES

Requires **LUCIDATA** Pascal ver 3.

**XREF** -- produce a Cross Reference Listing of any text; oriented to Pascal Source.

F and CCP - \$25.00

**INCLUDE** -- allows the inclusion of other files in a Source Text; has unlimited nesting capabilities. Also allows Binary File Inclusions.

F and CCP - \$25.00

**PROVILER** -- produces an Indented, Numbered, "Structogram" of a Pascal Source Text File. Allows viewing the overall structure of large programs, and provides clues as to the integrity of the program. Supplied as Source Code; requires compilation.

F and CCP - \$25.00

#### Lucidata

##### CCOPY

Pascal **NOT** required

Allows reading TSC Mini-FLEX, SSB DOS68, and Digital Research CP/M Disks while operating under FLEX 1.0, FLEX 2.0, or FLEX 9.0 with 6800 or 6809 Systems. **CCOPY** will not perform Miracles, but, between the program and the manual, you stand a good chance of accomplishing a transfer. Includes Utilities to List Directories, Copy Files, and convert Text Files when required. Also includes a Utility for investigating Physical Compatibility problems. Programs supplied in **MODULAR SOURCE** Code (Assembly Language) to make it easier to solve unusual problems.

F and CCP 5" - \$50.00  
 F 5" - \$65.00

#### Computer Systems Consultants

##### FLEX DISK UTILITIES

Eighteen (18) different FLEX Utilities that should be a part of every FLEX Users Toolbox. Ten BASIC Programs to:

Compare, Merge, or Generate Updates between two BASIC Programs, check BASIC Sequence Numbers, compare two unsequenced files, and 5 Programs for establishing a Master Directory of several Disks, and sorting, selecting, updating, and printing paginated listings of these files.

The other 8 Programs provide .OBJ Files which allow:

Copy a file with CRC errors, so it can possibly be salvaged; Test Disk for errors; Compare two Disks; a fast Disk Backup Program; Edit Disk Sectors; Monitor cross-chain on the Disk; print Disk Identification; and Sort and Replace the Disk Directory (in sorted order).

All Utilities include Source (either BASIC or Source Code). An EXCELLENT Value!

F and CCP - \$50.00

## BUSINESS

### WORD PROCESSORS

#### Alford and Associates

##### SCRIBITOR III

**EXTREMELY** Powerful Screen-Oriented Editor/Word Processor. Almost 50 different commands; **EXCELLENT** documentation (over 300 pages), including a full Tutorial Section to help you learn how to use the system. Features Cursor-based editing, dynamic Screen Formatting (what you see is what you get), Multi-column display and editing, "decimal align" columns (AND add them up automatically, if wanted), Define multiple keystroke macros, even and odd page number headers and footers, imbed printer control codes in text, full justification series of commands, full help support, store common command series on disk for future use, etc. Easy "Set-Up" (for example, you just hit the key you want to use for a specific function, such as "cursor up", and the System reads an stores that key - no digging into tech manuals for codes, etc.); use supplied "set-ups", or re-map the keyboard to what you are used too. Except for proportional printing, this package will DO IT ALL!

6800 or 6809 FLEX or SSB DOS, OS-9 - \$175.00

#### Great Plains Computer Co.

##### STYLICOMP

A full-screen oriented WORD PROCESSOR -- (now runs on the DataComp and FHL Color FLEX Systems; uses the 51 x 24 Display Screens). Full screen display and editing (i.e., what you see is what you get); supports the Daisy Wheel proportional printers.

SPECIAL CCP - \$195.00  
 F and O - \$295.00  
 U - \$395.00



\*FLEX is a trademark of Technical Systems Consultants  
 \*OS9 is a trademark of Microware

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 info (615) 842-4801

**CoCo OS-9™ FLEX™**  
**SOFTWARE**

#### Availability Legends --

F = FLEX, CCP = Color Computer FLEX  
 O = OS-9, COO = Color Computer OS-9  
 U = UNIFLEX  
 CCD = Color Computer Disk  
 COT = Color Computer Tape

# Great Plains Computer Co.

## MAIL MERCH

Greatly extends the power and flexibility of **STENOGRAPH**. Allows multiple Text files to be printed out as one large document. Provides for merging information into the Text File during printing (such as different names and addresses), etc.

F, CCF, O - \$145.00  
U - \$195.00

## Southeast Media

### SPELLS "Computer Dictionary"

OVER 120,000 words!

No more "let your fingers do the walking through the Dictionary" while you are entering Text with your favorite Editor or Word Processor. **SPELLS** is more than just "another Spelling Checker"; it allows you to look up a word from within your Editor or Word Processor so that you know it is right WHEN YOU TYPE IT IN with the **SPH.ORD** Utility (which operates in the **FLEX** Utility Space). Yes, it ALSO allows you to check and update the Text after you are finished; along with allowing you to ADD WORDS to the Dictionary, "Flag" questionable words in the Text for evaluation later, "View a word in context" before changing or ignoring, etc. **SPELLS** first checks a "Common Word Dictionary", then the normal Dictionary, then a "Personal Word List", and finally, any "Special Word List" you may have specified. **SPELLS** also allows the use of **Small Disk Storage** systems.

F and CCF - \$129.95

## Great Plains Computer Co.

### SPELL

Fast Computer Dictionary -- allows directly changing the Text File, adding words to the dictionary, etc. 75,000 words in less than 400 seconds.

F, CCF, OS/9 - \$125.00  
U - \$175.00

## DATA BASE MANAGEMENT SYSTEMS

### Watchtower Applied Business Systems

#### XIMS

Possibly one of the most powerful Database Management Systems available, this machine language program is small enough to operate on a single sided 5" disk, yet provides the speed of M.I. and power limited only by the user's imagination. This DMS supports Relational, Sequential, Hierarchical, and Random Access File Structures, and has Virtual Memory capabilities for those Giant Data Bases. **XIMS Level I** provides a functional "entry level" System which provides for defining a Data Base, entering and changing the Data, and producing Reports. **XIMS Level II** adds the POWERFUL "QUERY" facility which uses an English Language Command Structure in manipulating the Data to create new file Structures, Sort, Select, Calculate, etc. **XIMS Level III** adds several special "Utilities" which provide additional ease of working with the various structures, changing System Parameters, etc.

**XIMS Lvl I** - F & CCF - \$129.95  
**XIMS Lvl II** - F & CCF - \$199.95  
**XIMS Lvl III** - F & CCF - \$269.95  
**XIMS System Manual** only - \$24.95

## Great Plains Computer Co.

### FORMING DBMS

An **XBASIC**, Menu Driven, DBMS with "Built-in" Audit Tracking, Extremely Powerful Report & Format Capabilities, etc. This **Time Proven** DBMS will become the "Work Horse" of your Software Stable.

F and CCF - \$295.00  
U - \$395.00

## ACCOUNTING PACKAGES

### Great Plains Computer Co.

#### Accounting Packages

**Accts Rec., Accts Payable & Gen Ledger** -- A **FULL** Accounting Package that can be used together, or as separate packages; provides the IRS required Audit Tracking. (**XBASIC**, based on the "Deborne Business Programs.")

F and CCF - ea. Program \$295.00  
U - ea. Program \$395.00



\*FLEX is a trademark of Technical Systems Consultants  
\*OS9 is a trademark of Microware

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343

info (615) 842-4601

**CoCo OS-9" FLEX" SOFTWARE**

**TOLL FREE**  
**1-800-338-6800**  
For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
for information  
call (615) 842-4601

**CoCo OS-9" FLEX" SOFTWARE**

## Universal Data Research, Inc.

### Accounting and Database Manag. Sys.

Note: ALL UDRI Accounting and DBM Progs. Require **FLEX** and **XBASIC**. These are Time Tested programs from an old, established, software house.

## Color Computer FLEX Systems

DBM Part 1 - \$49.95	DBM Part 2 - \$49.95
Church Contributions - \$49.95	Single Entry Gen Ledger - \$49.95
Balanced Billing System - \$49.95	Gen Ledger - \$189.00
A/R \$99.95	Inventory 2 \$69.00
A/P \$99.95	Payroll \$99.95

## FLEX and UniFLEX -- Notes: Requires XBASIC or basic

A/P	F - \$295, U - \$395
A/R	F - \$295, U - \$395
Gen Ledger	F - \$295, U - \$395
Inventory 2	F - \$295, U - \$395
Payroll	F - \$295, U - \$395
DBM	F - \$350, U - \$450

## Computer Systems Consultants

### FULL SCREEN INVENTORY/HRP

The Full Screen Inventory System provides a means of maintaining small inventories. Using a linked, keyed random file structure based upon the item field, it keeps the file in alphabetical order for easier inquiry. With the FIND command, the user may locate and/or print all records matching on partial or complete item, description, vendor, or attributes. Items in backorder or below minimum stock levels may be located and/or printed thru the same process. Printed output may be produced in item or vendor order. A materials requirement planning (MRP) capability for manufacturing environments is included to allow the maintenance and analysis of Hierarchical assemblies of items in the inventory file. It requires TSC's **Extended BASIC**.

F and CCF - \$180.00, U - \$150.00

## BUSINESS FORECASTING

### The Virginia Company

#### Bizpack

**BIZPACK** is used for storing accounting, numeric, and financial data which can then be used for planning, budgeting, forecasting, analyzing, etc. While "Electronic Spreadsheets" are extremely useful in many situations, **BIZPACK** excels in businesses where there are numerous expense columns, revenue sources, significant business indicators, large numbers, erratic week-to-week and month-to-month fluctuations, etc. **BIZPACK** helps determine statistical relationships, establish trend lines, "smooths" data via moving averages, analyze seasonal data, adjusts for inflation, lags data in Statistics or Column functions, plots data, etc. **BIZPACK** is oriented toward time series analysis of businesses. The Program displays information on the screen in Columns of information with each Row conforming to a defined Period of Time (weeks, months, years, etc.), and is very easy to use (data is easy to enter, change, and modify; commands can be renamed to suit the users requirements; unlimited ability to create specialized commands using common BASIC Statements; etc.). Requires TSC's **Extended BASIC**.

F and CCF - \$135.00  
with Source - \$250.00

## Availability Legends --

F = FLEX, CCF = Color Computer FLEX  
O = OS-9, CCF = Color Computer OS-9  
U = UniFLEX  
ODD = Color Computer Disk  
CTT = Color Computer Tape

**TOLL FREE**  
**1-800-338-6800**  
 For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 for information  
 call (615) 642-4601

**CoCo OS-9™ FLEX™**  
**SOFTWARE**

**SPECIAL**  
 Purchase X-BASIC and X-BASIC together for \$221.50  
 — a Savings of \$13.50 —

#### Computer Systems Consultants

##### TABULA RASA SPREADSHEET

TABULA RASA is similar to DESKTOP/PLAN and provides for the generation and maintenance of tabular computation schemes often used for analysis of business, sales, and economic scenarios. Its menu-driven user interface provides these capabilities even to those users with no programming experience. Its extensive report-generation capabilities allow the user to generate professional results with minimum effort. It requires TSC's Extended BASIC.

F and CCP = \$189.00, U = \$208.00

#### Computer Systems Center

##### OMNICALC

THE Electronic Spread Sheet for 6809 Computer Systems. An extremely POWERFUL Business Tool, this Program will find an unlimited number of "non-business" applications, also (for example, a Full Junior College Electronics Curriculum was set up using OMNICALC). Advanced features like "Table Lookup" make Income Tax work easy; Column or Row Sorting for numerous applications; etc. Completely "Memory Resident", Machine Language, this Program is FAST. Provides STANDARD FLEX Text File output for use with BASIC, Word Processors, Pascal, C, etc. Also available for Data-Corp and FHL FLEX systems using the 50 x 24 Displays.

F and SPECIAL CCP = \$289.00  
 U = \$395.00

### ODDS & ENDS

#### Computer Systems Consultants

##### FULL SCREEN FORM DISPLAY

This package supports any Serial Terminal with cursor control of Memory-Mapped Video Displays. The package substantially extends the screen input/output capabilities of TSC's Extended BASIC programs by providing a simple, table-driven method of describing and using full screen displays. These table entries are easy to set up and maintain, and are normally stored on disk and read as required. A simple, interactive means of generating the forms and the data field definitions is provided.

F and CCP = \$59.00, U = \$75.00

#### Computer Systems Consultants

##### FULL SCREEN MAILING LIST

The Full Screen Mailing List System provides a means of maintaining simple mailing lists. Using a random fill structure based on the first character of the name field, it maintains the file in alphabetical order for easier inquiry. With the FIND command, the user may locate all records matching on partial or complete name, city, state, zip, or attributes. Printed listings and output to labels may also be produced on the same selective basis. It requires TSC's Extended BASIC.

F and CCP = \$189.00, U = \$119.00

### COLOR COMPUTER SOFTWARE

#### Stearns Electronics

##### FORTH

Intrigued by FORTH?? Here is a FORTH package tailored to the Color Computer! This package is supplied on Tape, with instructions for transferring it to disk if you wish. Written primarily in machine language, it's speed is unparalleled. A full Semigraphic-8 editor is provided, along with "gadgets" like Graphics and Sound Commands, Printer Commands, Auto-Repeat and Control Keys, etc. If you are interested in Learning FORTH, a Trace feature is provided which is invaluable. If you are a FORTH Pro, this package provides CPU carry flag accessibility, Fast Task Multiplexing, Clean Interrupt Handling, etc. (Or; you won't "out grow" the Basic capabilities of this Implementation). Combine this package with Leo Brodie's EXCELLENT book "Starting FORTH", and you will be a FORTH Expert before you know it (and have a lot of fun doing it!).

Color Computer TAPE = \$58.95

#### Custom Software Engineering, Inc.

##### Color Computer GRAPHIC SCREEN PRINT Programs

Dumps any "PMODE" Screen to the Printer with the BASIC USR Function. Shift the Printout Left or Right or Reverse Print (Dark for Light Screen and Vice Versa). All Programs on Tape.

CCSPR for R.S. LP-VII/VIII & IMP 100/200/400 \$7.95  
 CCSPZ for Epson w/ Craftax and Craftax + \$9.95  
 CCSPG for Gemini 10 and 15 \$9.95  
 CCSPD for the Prowriter Printers \$9.95

#### Custom Software Engineering, Inc.

##### DATE-O-BASE CALENDAR Program

A Menu Driven EXTENDED BASIC Program which allows the entry of up to 12 Memos per Day, each of which may contain up to 28 Characters, for any day of the Month between the years 1700 and 2099. A Graphic Calendar shows which days contain Memos, and a "Key Word" Search is provided which can be output to the Screen or Printer.

TAPE DATE-O-BASE CALENDAR  
 (Each Tape File will hold up to 400 Memos) \$16.95  
 DISK DATE-O-BASE CALENDAR  
 (4,000 Memos at 300/Month per Disk) \$19.95

#### Custom Software Engineering, Inc.

##### That's INTEREST

Interested in INTEREST (the Money Kind)? An EXTENDED BASIC Program that will help you deal with numerous problems requiring Interest calculations. Present Value, Rate of Return, Current Bond Yield and Rate of Return to maturity, Loan Repayment Amortization Schedules, etc.

TAPE = \$29.95

#### Custom Software Engineering, Inc.

##### DISK DATA CHAINED FILE

An EXTENDED BASIC Data Management System w/ Mach. Lang. Routines. Allows a max of 246 Chars. and 14 Fields per Record, and another Record can be linked to the first; 8 Char. Field Names, up to 99 Chars. per Field. Powerful On-Screen editor for input and update, flexible Output capabilities including output to Disk Files for use by other Programs. Change File Definition without re-entering the Data, Split Files, etc. Allows Multiple Field Sorts, Select on any combination of Fields, etc. An extremely POWERFUL TOOL! Instructions provide examples of Mailing Lists and a Financial Stock Profit and Loss Tracking System.

DISK = \$54.95

#### Custom Software Engineering, Inc.

##### DISK DOUBLE ENTRY

DISK EXTENDED BASIC Accounting Program w/ Mach. Lang. Routines. A "Traditional" Accounting Package for Small Business, Clubs, Churches, Personal Use, etc. Up to four levels of subtotals with Trial Balance, Income Statement, and Balance Sheet Reports. NOE allows up to 300 accounts and a Trial Balance of \$9,999,999.99. Transactions may be up to 14 lines long, and comments and explanations may be freely used. Accounts are traceable to the journal transaction, which may include comments. Screen reports allow review of past transactions and current balances.

DISK = \$44.95



"FLEX" is a trademark of Technical Systems Consultants  
 "OS9" is a trademark of Microware

**TOLL FREE**  
**1-800-338-6800**  
 For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 info (615) 642-4601

**CoCo OS-9™ FLEX™**  
**SOFTWARE**

#### Availability Legend —

F = FLEX, CCP = Color Computer FLEX  
 O = OS-9, CDD = Color Computer OS-9  
 U = UnifLEX  
 CDD = Color Computer Disk  
 COT = Color Computer Tape

**OS/9**-- Multi-User, Multi-Tasking with **FLEX** --

**Southeast Media** is now shipping **OS/9** FROM STOCK - the multi-user, multi-tasking capability of **OS/9** allows **FLEX** users the advantages of more sophisticated and time saving computer usage without having to buy or learn a new language or operating system syntax. **OS/9**, as its name implies, allows true "time-sharing" operation under the popular **FLEX** operating system, and also allows each user to run two simultaneous jobs (multi-tasking); even on single-user systems. For example, while in **EDIT**, you can list another file or examine a directory. Or, you might look up an item in a Data Base while a Sort is in progress! **OS/9** also provides some fringe benefits that will be greatly appreciated by **FLEX** users, including type-ahead, command line editing, and instant response to "escape".

**OS/9** is the painless method! Use your existing **FLEX** computer by simply adding 64K of RAM for each user and/or task. Fact is, you still use **FLEX** just like you always have! **OS/9** is not intended as competition to **UNIX**. It does not improve on the speed of **FLEX**, and does not offer password protection or other niceties of a full-blown multi-user system. What **OS/9** does do is give **FLEX** users a **low-cost** way to use existing software in a multi-user, multi-tasking environment, so your existing **FLEX** versions of **BASIC**, **XBASIC**, editors, assemblers, disassemblers, sort/merge packages, word processors, compilers, **OS/9** spread-sheet package, and so on are still good.

**NOTE** -- The initial release of **OS/9** is for **8086** 8/09 Computers, but versions will also be available for other popular extended-memory (up to **16MB**) systems, such as **HELD** and **COMX**. A minimum of **128K** of RAM will be required with **ALL** versions. **OS/9** requires **64K** of RAM for each active task; thus a **256K** system could allow foreground-background operation on two terminals, or foreground-only operation on four terminals.

AVAILABLE NOW from Southeast Media - \$289.00

**AUTHORS - PROGRAMMERS****QUALITY SOFTWARE NEEDED****FLEX** - **UniFLEX** - **OS/9** - Color Computer

For the past several months, we at the **Southeast Media Division of Computer Publishing, Inc. (CPI)**, the parent company of '68' **MICRO JOURNAL** and **COLOR MICRO JOURNAL**, have debated expanding our software distribution business. Many other magazines have been doing so for years (in fact, **MOST** were in the Software Distribution Business **BEFORE** they began to publish a Magazine). Presently there are many fine examples of software that has been developed by **YOU**, our readers, that will never see the "light of day" due to the **Cost of Advertising and TIME and Cost** involved in the production, distribution, and **Customer SUPPORT** of that software unless **SOMEONE**, with enough exposure and the willingness to continually advertise, runs with the ball.

Software is the "backbone" for the **REAL** utilization of any Computer System, and ours are no exception! This has been no simple decision. While we realize that there could be some conflict with some of our advertisers, we **ALSO** hear a **LOUD** and **CONTINUOUS** cry for **HELP** from our Readers. From day one, the **foremost** concern of '68' **MICRO JOURNAL** has been it's **READERS!** Therefore, our **Southeast Media Division** will accept, for appraisal for possible Distribution, **6809** software; Games, Utilities, Software Development, Business Application Programs, etc.



\***FLEX** is a trademark of Technical Systems Consultants  
 \***OS/9** is a trademark of Microware

**TOLL FREE**  
**1-800-338-6800**  
 For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 info (615) 842-4801

**CoCo OS-9" FLEX" SOFTWARE**

**Availability Legends** --

**F** = **FLEX**, **CCP** = Color Computer **FLEX**  
**O** = **OS-9**, **CCD** = Color Computer **OS-9**  
**U** = **UniFLEX**  
**CCD** = Color Computer Disk  
**CCP** = Color Computer Tape

**TOLL FREE**  
**1-800-338-6800**  
 For Ordering

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 for information  
 call (615) 842-4801

**CoCo OS-9" FLEX" SOFTWARE**

In the past there has been too much software offered that was not quite ready. We will strive to eliminate that element. But, right up front, we tell you only that we will do our very best; nothing more. Also, we will strive to keep cost to a bare minimum, while securing for the author a fair return in royalty payments, promptly paid, and in customer support for his product.

Of course, we will expect, no -- **DEMAND**, that the author keep the product free of errors (bugs), and maintain it in a prompt and business like manner. Also we shall require that authors be willing to furnish 'source' for those programs that justify, by price and utility, inclusion of same. The lack of source code, properly commented, is a continual complaint we hear. Not all programs will be sold with source, but where necessary, we will insist that it be included.

In some instances the program may be small or short and not justify itself as a "single" sale product. In this event it will be combined with other like programs, and offered as a package. In that event, the royalties will be split between the various authors.

If you have software that you feel will qualify under this program, please contact one of the people below. Remember, if your software has any problems or "funnies" -- **GET IT STRAIGHT BEFORE YOU CONTACT US!** Also get your source code in proper shape and well commented; there is too much 99% code already drifting around.

If your software is **READY** contact:  
 Bob May, Don Williams, or Tom Williams

**Southeast Media** is a division  
 of **Computer Publishing, Inc. (CPI)**,  
 a family of 100% **680X** support facilities.





# TEN MOST-ASKED QUESTIONS about **DYNACALC**<sup>TM</sup>

## THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

---

**1. What is an electronic spread-sheet, anyway?**

Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

**2. Is DYNACALC just for accountants, then?**

Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. You can build worksheets as large as 256 columns or 256 rows. There's even a built-in sort command, so you can use DYNACALC to manage small data bases — up to 256 records.

**3. What will DYNACALC do for ME?**

That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc<sup>TM</sup>, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications. Typical uses include financial planning and budgeting, sales records, bills of material, depreciation schedules, student grade records, job costing, income tax preparation, checkbook balancing, parts inventories, and payroll. But there is no limit to what YOU can do with DYNACALC.

**4. Do I have to learn computer programming?**

NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Even experienced programmers can get jobs done many times faster with DYNACALC, compared to conventional programming. Built-in HELP messages are provided for quick reference to operating instructions.

**5. Do I have to modify my system to use DYNACALC?**

Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

**6. Will DYNACALC read my existing data files?**

You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, STYLOGRAPH<sup>TM</sup> word processor, RMS<sup>TM</sup> data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

**7. How fast is DYNACALC?**

Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

**8. Is there a version of DYNACALC for MY system?**

Probably. You need a 6809 computer (32k minimum) with FLEX<sup>TM</sup>, UNIFLEX<sup>TM</sup>, or OS-9<sup>TM</sup> operating system. You also need a decent crt terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UNIFLEX and OS-9 versions of DYNACALC allow you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX (Frank Hogg or Data-Comp versions).

**9. How much does DYNACALC cost?**

The FLEX versions are just \$200 per copy; UNIFLEX version \$395; OS-9 version (works with LEVEL ONE or LEVEL TWO) \$250. Orders outside North America add \$7 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.

**10. Where do I order DYNACALC?**

See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX or OS-9 versions. Software serial number is required for the UNIFLEX version.

---

### Order your **DYNACALC** today!

---

**Foreign Dealers:**

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

---

**Computer Systems Center**  
13461 Olive Blvd.  
Chesterfield, MO 63017  
(314) 576-5020



---

UNIFLEX software prices include maintenance for the first year.

**DYNACALC is a trademark of  
Computer Systems Center**

VisiCalc is a trademark of VisiCorp.  
STYLOGRAPH is a trademark of Great Plains Computer Co.  
RMS is a trademark of Washington Computer Services.  
FLEX and UNIFLEX are trademarks of TSC.  
OS-9 is a trademark of Microware and Motorola.

# DYNAMITE+™

## "THE CODE BUSTER"

disassembles any 6809 or 6800 machine code program into beautiful source

- Learn to program like the experts!
- Adapt existing programs to your needs!
- Convert your 6800 programs to 6809!
- Automatic LABEL generation.
- Allows specifying FCB's, FCC's, FDB's, etc.
- Constants input from DISK or CONSOLE.
- Automatically uses system variable NAMES.
- Output to console, printer, or disk file.
- Available for all popular 6809 operating systems.

FLEX™ \$100 per copy; specify 5" or 8" diskette.

OS-9™ \$150 per copy; specify 5" or 8" diskette.

UNIFLEX™ \$300 per copy; 8" diskette only.

For a free sample disassembly that'll convince you DYNAMITE+ is the world's best disassembler, send us your name, address, and the name of your operating system.

### Order your DYNAMITE+ today!

See your local DYNAMITE+ dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Orders outside North America add \$5 per copy. Please specify diskette size for FLEX or OS-9 versions.

### Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

**Computer Systems Center**  
13461 Olive Blvd.  
Chesterfield, MO 63017  
(314) 576-5020



UNIFLEX software prices include maintenance for the first year.

DYNAMITE+ is a trademark of Computer Systems Center.

FLEX and UNIFLEX are trademarks of TSC.  
OS-9 is a trademark of Microware and Motorola.  
Dealer inquiries welcome.

## '68' MICRO JOURNAL

- ★ The only ALL 6800 Computer Magazine.
- ★ More 6800 material than all the others combined:

### MAGAZINE COMPARISON (2 years)

#### Monthly Averages

KB	BYTE	6800 Articles		TOTAL PAGES
		CC	DOBB'S	
7.8	6.4	2.7	2.2	19.1 ea. mo.

Average cost for all four each month: \$6.53  
(Based on advertised 1-year subscription price)

'68' cost per month: \$2.04

That's Right! Much. Much More

for About

1/3 the Cost!

OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card # \_\_\_\_\_ Exp. Date \_\_\_\_\_

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: \$ \_\_\_\_\_

Name \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

My Computer Is: \_\_\_\_\_

68 Micro Journal  
6900 Cassandra Smith Rd.  
Hixson, TN 37343

### SUBSCRIPTION RATES

#### USA

1 Year \$24.50, 2 Year \$42.50, 3 Year \$64.50

\*FOREIGN SURFACE Add \$12.00 per Year to USA Price

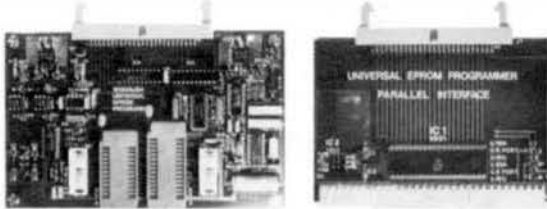
\*FOREIGN AIRMAIL Add \$36.00 per Year to USA Price

\*\*CANADA & MEXICO Add \$5.50 per Year to USA Price  
Cash (USA) or drawn on a USA Bank!!!



# WINDRUSH MICRO SYSTEMS

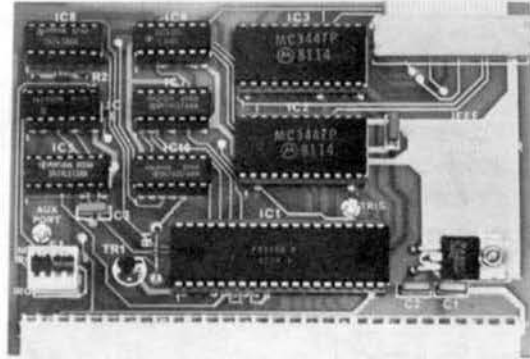
## UNIVERSAL EPROM PROGRAMMER



- PROGRAMS and VERIFIES 2508, 2708, 2516, 2716, 2532, 2732A, 2564, and 2764 EPROMS. Minor hardware mods are required to program the INTEL 27128.
- Tri-volt and Single Volt 2508/2708 and 2516/2716 devices are supported.
- ZIF sockets with mode selector switches eliminate 'personality modules'.
- Twin boards with five feet of twisted pair planar cable puts the programmer out on the bench where it belongs.
- SS-30 and EXORCISOR interfaces are available.
- Menu driven software provides the following facilities:
  - a. MOVE blocks of memory within the buffer.
  - b. READ an EPROM into the buffer.
  - c. VERIFY an EPROM against the buffer.
  - d. EXAMINE and change the contents of the buffer.
  - e. PUMP the contents of the buffer in HEX and ASCII.
  - f. FILL a selected area of the buffer with a specified character.
- Software available for all versions of SS8 DOS, FLEX 2, FLEX 9 and OS-9. Assembly language source files supplied on disk....enables customising.
- Well documented users manual provides step-by-step adaptation and operating instructions.

AVAILABLE FROM GIMIX IN THE U.S.A.

## IEEE-488



- SUPPORTS ALL PRINCIPAL MODES OF THE IEEE-488 (1975/8) BUS SPECIFICATION:
  - Talker
  - Listener
  - System Controller
  - Serial Poll
  - Parallel Poll
  - Group Trigger
  - Single or Dual Primary Address
  - Secondary Address
  - Talk only...Listen only
- Fully documented with a complete reprint of the KILBAUD article on the IEEE bus.
- Low level assembly language drivers suitable for 6800, 6801, 6802, 6803, 6808 and 6809 are supplied in the form of listings. These drivers have been extensively tested and are GUARANTEED to work!
- Single SS-30 board (4, 8, or 16 addresses per port), fully socketed, gold plated bus connectors, and IEEE interface cable assembly.

## PL/9 EDITOR/COMPILER/DE-BUGGER

- Friendly inter-active environment where you have INSTANT access to the Editor, the Compiler, and the Trace-debugger, which, amongst other things, can single step the program a SOURCE line at a time. You also have direct access to any FLEX utility and your System Monitor.
  - 250 page manual is organized as a tutorial with plenty of examples.
  - Fast single pass compiler produces 8K of COMPACT and FAST 6809 machine code output per minute with no run-time overheads or license fees.
  - Fully compatible with TSC text editor format disk files.
  - Signed and unsigned BYTES and INTEGERS, 32-bit floating point REALs.
  - Vectors (single dimension arrays) and Pointers are supported.
  - Mathematical expressions: (+), (-), (\*), (/), modulus (%), negation (~)
  - Expression evaluators: (<+>), (<+>), (<+>), (<+>), (<+>)
  - Bit operators: (<AND>), (<OR>), (<EOR/XOR>), (<NOT>), (<SHIFT>), (<SWAP>)
  - Logical operators: (<AND>), (<OR>), (<EOR/XOR>)
  - Control statements: IF..THEN..ELSE, IF..CASE1..CASE2..ELSE, BEGIN..END, WHILE.., REPEAT..UNTIL, REPEAT..FOREVER, CALL, JUMP, RETURN, BREAK, GOTO.
  - Direct access to (ACCA), (ACCB), (ACCD), (CC) and (XREG).
  - FULLY supports the MC6809 SWI, SWI2, SWI3, NMI, FIRQ, IRQ and RRETR vectors. Writing a self-starting (free power-up) program that uses ANY, or ALL, of the MC6809 interrupts is an absolute snap!
  - Procedures may be passed and may return variables. This makes them functions which behave as though they were an integral part of PL/9.
  - Several fully documented library function modules are supplied: IOSUBS, BITIO, HARDIO, MEXIO, FLEXIO, SCIPACK, STRSUBS, and REALCOM.
- '... THIS IS THE MOST EFFICIENT COMPILER I HAVE FOUND TO DATE.'

Quoted from Ron Anderson's FLEX User Notes column. Need we say more?

WE STOCK THE FOLLOWING COMPANIES PRODUCTS:  
GIMIX, SS8, FHL, MICROWARE, TSC, LUCIDATA, AND ALFORD & ASSOCIATES.

FLEX (tm) is a trademark of Technical Systems Consultants, OS-9 (tm) is a trademark of Microware Systems Corporation, MODS (tm) and EXORCISOR (tm) are trademarks of Motorola Incorporated.

## MACE/XMACE

A co-resident EDITOR/ASSEMBLER for the 6809 written by Graham Trott which takes most of the pain out of assembly language program development:

- Friendly inter-active environment where you have INSTANT access to the Editor, the Assembler, FLEX and your System Monitor.
- MACE can also produce ASMPPDC's for PL/9 with the assembly language source pressed to the output file as comments.
- Includes XMACE a co-resident 6800/1/2/3/8 EDITOR/CROSS ASSEMBLER.

**C**

This is the FLEX version of the James Mcintosh 'C' compiler that is also available on UNIFLEX from SWTP and OS-9 from Microware:

- The FLEX implementation supports the full Kernighan and Ritchie 'C' specification except 'floats', 'doubles', and 'bit-fields'.
- Produces very efficient assembly language source output with the 'C' source optionally interleaved as comments.
- Built-in optimizer will shorten object code by about 11%
- Supports interleaved assembly language programs.
- The TSC relocating assembler/linking loader (SP09-17) is REQUIRED.

MACE	(6809 FLEX only).....	\$ 98.00
PL/9	(6809 FLEX only)....steal at this price!.....	\$198.00
'C'	(a 56K system and the TSC SP09-17 package is req'd)....	\$293.00
IEEE-488	with IEEE-488 cable assembly.....	\$298.00
SS-30	Universal EPROM Programmer w/one version of software...	\$375.00
EXORCISOR	Universal EPROM programmer w/one version of software...	\$395.00
SOFTWARE	Drivers for a 2nd, 3rd, or 4th operating system.....	\$ 25.00

ALL PRICES INCLUDE AIR MAIL POSTAGE

An SS-30C all CMOS 256K STATIC RAM board will be available SOON!

Write for details & pricing.

WORSTED LABORATORIES,  
NORTH WALSHAM, NORFOLK,  
ENGLAND. NR28 9SA.  
TEL: (0692) 405189  
TLX: 97360 SHARET G

FEATURES THE  
POWERFUL, THIRD  
GENERATION,  
MOTOROLA 6809  
PROCESSOR!

## THE 6809 "UNIBOARD"<sup>™</sup> SINGLE BOARD COMPUTER KIT

PERFECT FOR COLLEGES, OEM'S, INDUSTRIAL  
AND SCIENTIFIC USES!

64K RAM! DOUBLE DENSITY  
FLOPPY DISK CONTROLLER!

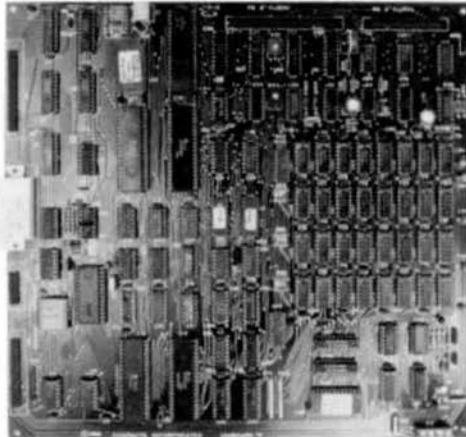
*New!*

BLANK PC BOARD

**\$99<sup>95</sup>**

WITH PAL'S, AND  
TWO EPROMS.

FOR 5-1/4 OR 8 INCH  
SOURCE DISKETTE  
ADD \$10.



**\$399<sup>00</sup>**

COMPLETE KIT!  
FULLY SOCKETED.

ALL OPTIONS ARE  
STANDARD. NO  
EXTRAS TO BUY!

**THE COMPACTA UNIBOARD<sup>™</sup>:** Through special arrangement with COMPACTA INC., we are proud to have been selected the exclusive U.S. Mfg. of their new 6809 UNIBOARD<sup>™</sup> COMPUTER KIT. Many software professionals feel that the 6809 features probably the most powerful instruction set available today on ANY 8 bit micro. Now, at last, all of that immense computing power is available at a truly unbelievably low price.

### FEATURES:

- \* 64K RAM using 4116 RAMS.
- \* 6809E Motorola CPU.
- \* Double Density Floppy Disk Controller for either 5-1/4 or 8 inch drives. Uses WD1793.
- \* On board 80 x 24 video for a low cost console. Uses 2716 Char. Gen. Programmable Formats. Uses 6845 CRT Controller.
- \* ASCII keyboard parallel input interface. (6522)
- \* Serial I/O (6551) for RS232C or 20 MA loop.
- \* Centronics compatible parallel printer interface. (6522)
- \* Buss expansion interface with DMA channel. (6844)
- \* Dual timer for real time clock application.
- \* Powerful on board system monitor (2732). Features commands such as Go To, Alter, Fill, Move, Display, or Test Memory. Also Read and Write Sectors. Boot Normal, Unknown, and General Flex<sup>™</sup>.

### YOUR CHOICE OF POPULAR DISK OPERATING SYSTEMS:

FLEX <sup>™</sup> from TSC	\$149
OS9 <sup>™</sup> from Microware	\$199
Specify 5-1/4 or 8 Inch	

PC BOARD IS  
DOUBLE SIDED, PLATED THRU  
SOLDER MASKED. 11 x 11-1/2 IN.

**Digital Research Computers**  
(OF TEXAS)

P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 271-3538

**TERMS:** Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. Add \$4.00 shipping. USA AND CANADA ONLY

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY  
LIMITED WARRANTY. A FREE COPY IS AVAILABLE UPON REQUEST.



# 64K SS-50 STATIC RAM

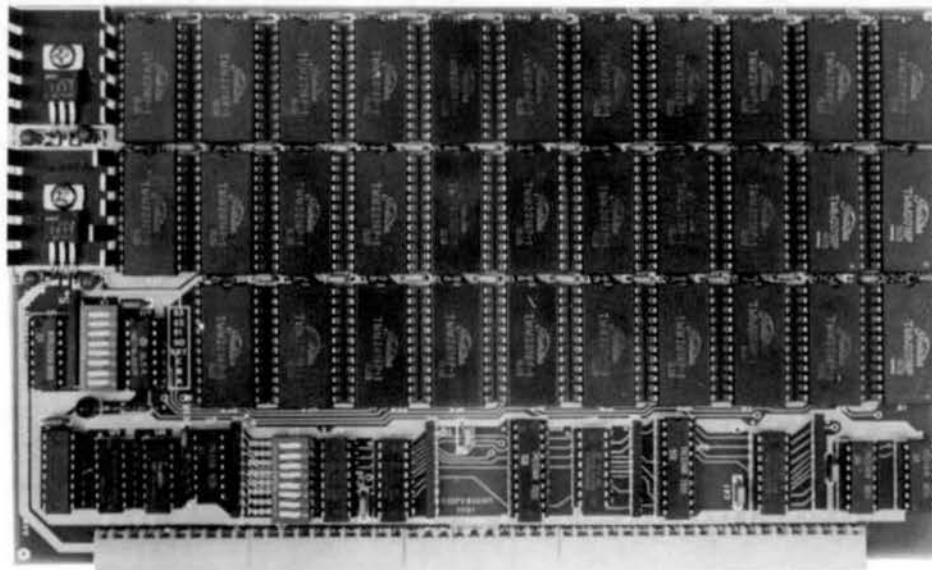
**\$179<sup>00</sup>**  
(48K KIT)

**NEW!**

**NEW!**

**LOW  
POWER!**

**RAM  
OR  
EPROM!**



**BLANK PC BOARD  
WITH DOCUMENTATION  
\$52**

**SUPPORT ICs + CAPS - \$18.00  
FULL SOCKET SET - \$15.00**

**ASSEMBLED AND TESTED ADD \$50**

## FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

<b>56K</b>	<b>\$219</b>
<b>64K</b>	<b>\$249</b>

## 16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

**CLOSE OUT SPECIAL**  
WE HAVE DROPPED OUR 32K SS-50 STATIC RAM BOARD WHICH USED 2114 LOW POWER RAMS. WE WILL SELL THE REMAINING STOCK OF BLANK PCB'S WITH DATA FOR \$17.50 EA. THESE FORMERLY SOLD FOR \$50.

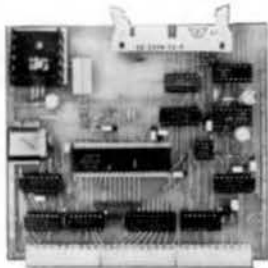
**Digital Research Computers**  
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

**TERMS:** Add \$2.00 postage. We pay balance. Order under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50, add 85¢ for insurance.

## DOUBLE DENSITY OPERATION FOR 6800 AND 6809 COMPUTERS

**DOUBLE  
YOUR  
STORAGE  
CAPACITY**



**SS-30  
or  
SS-30C**

Now you can run double density on any S-50 6800 or 6809 computer. Who else can match these features at these low prices! The FD-2 features:

- \* Control up to four 5 1/4" DS/DD Drives
- \* SS-30 or SS-30C compatible
- \* Uses the TSC or SWTPC FLEX Operating System
- \* 20 MHz operation with no "slow I/O required"
- \* Run double density on 6800 systems having a 894 KHz clock or faster
- \* Compatible with SWTPC DC1, DC2, DC3, or DC4 controllers

FD-2	Assembled and tested controller card	\$149.95
FD-2S	6800 Double Density Drivers + Format Program	\$ 19.95
FD-2SDDD	FD-2, 2 40 TR Double Sided Drives, Cabinet	
	Power Supply, and Cables	\$749.95
FD-1A	CLOSE OUT SPECIAL! FD-1 Single Density Controller	\$ 49.95

### PERIPHERAL TECHNOLOGY

"Supplying Floppy Disk Controllers Since 1978"

3760 Lower Roswell Road  
Marietta, Georgia 30067

VISA/MASTERCARD/CHECK/COD 404/973-0042

<sup>TM</sup>FLEX is a trademark of Technical Systems Consultants

## MACROPLEX Software

175 Fifth Avenue, Suite 3011, New York, NY 10010

### Presenting... **THE CONDUIT** A new File Type for OS9

**THE CONDUIT** lets programs exchange data without using external files. Direct program-to-program data transfers on any path use OS9 I/O calls. No need to change device-independent programs. Two-way transfers are permitted and device-dependent code can be accommodated. Programs that formerly ran sequentially can now run concurrently to the limit of memory.

Utilities supplied with **THE CONDUIT** give immediate results when linked to your current programs via **CONDUIT** paths:

- Fast disk store-down speeds up assemblies and other I/O bound procedures.
- "In-flight" terminal I/O redirection (e.g., set traps in DEBUG from a disk file).
- Basic OS9 source inclusions, enhanced listing.
- "Null files" drain output, signal EOF on input.

Free brochure available. Write, or call (212) 686-3036.

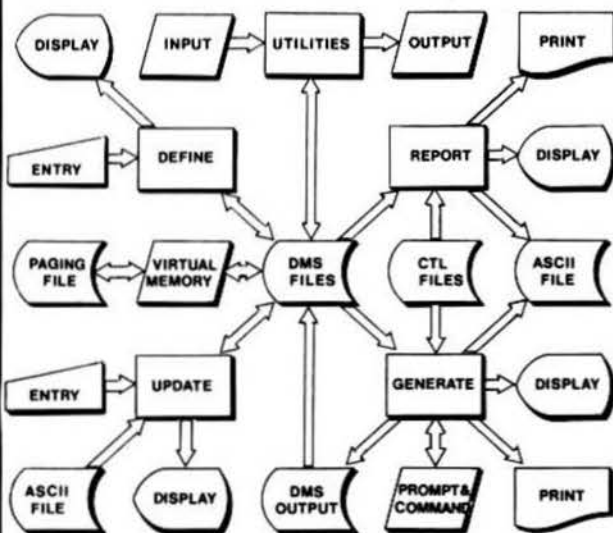
Manual alone (50 pages)	\$ 8
CONDUIT file system + utilities	\$70
Both together	\$75
Postage outside North America, add	\$ 5
Specify 5" or 8" disk	

Version for **OS9 Level Two** available soon

OS9<sup>TM</sup> Basic OS<sup>TM</sup> Microware Systems Corp. & Motorola, Inc.

# XDMS

## Data Management System



System Architecture

**WESTCHESTER Applied Business Systems**  
Post Office Box 187  
Briarcliff Manor, N.Y. 10510

### XDMS Data Management System

The XDMS Data Management System is available in three levels. Each level includes the XDMS nucleus, VMOEM utility and System Documentation for level III. XDMS is one of the most powerful systems available for 6809 computers and may be used for a wide variety of applications. XDMS users are registered in our database to permit distribution of product announcements and validation of user upgrades and maintenance requests.

#### XDMS Level I

XDMS Level I consists of DEFINE, UPDATE and REPORT facilities. This level is intended as an "entry level" system, and permits entry and reporting of data on a "tabular" basis. The REPORT facility supports record and field selection, field merge, sorting, line calculations, column totals and report titling. Control is via a English-like language which is upward compatible with level II. XDMS Level I . . . . \$129.95

#### XDMS Level II

Level II adds to Level I the powerful GENERATE facility. This facility can be thought of as a general file processor which can produce reports, forms and form letters as well as file output which may be re-input to the facility. GENERATE may be used in complex processing applications and is controlled by a English-like command language which encompasses that used by Level I. XDMS Level II . . . . \$149.95

#### XDMS Level III

Level III includes all of level II plus a set of useful DMS Utilities. These utilities are designed to aid in the development and maintenance of user applications and permit modification of XDMS system parameters, input and output of XDMS files, display and modification of file format, graphic display of numerical data and other functions. Level III is not need for advanced XDMS users. XDMS Level III . . . . \$269.95  
XDMS System Documentation only (\$10. credit toward purchase). . . \$ 24.95

### XACC Accounting System

The XACC General Accounting System is designed for small business environments of up to 10,000 accounts and inventory items. The system integrates accounting functions and inventory plus the general ledger, accounts receivable and payable functions normally sold separately in other systems. Features user defined accounts, products (or services), transactions, invoicing, etc. Easily configured to most environments. XACC General Accounting System (Requires XDMS, pref. Lv. III). . . \$299.95  
XACC System Documentation only (\$10. credit toward purchase). . . \$ 24.95

**WESTCHESTER Applied Business Systems**  
Post Office Box 187, Briarcliff Manor, N.Y. 10510

All software is written in macro/assembly and runs under 6809 FLEX O/S. Terms: Check, Money Order, Visa or Mastercard. Shipment first class. Add P&H \$2.50 (\$7.50 Foreign). NY Res add sales tax. Specify 5" or 8".

Sales: S. E. MEDIA, 1-800-336-6800, Consultation: 914-941-3552 (evening).

FLEX is a trademark of Technical Systems Consultants, Inc.

## 6809 COMPLETE SYSTEM 6809 ADVANCED TECHNOLOGY ECONOMICAL



OEM Inquiries  
Welcome

The PT69 COMPLETE SYSTEM features the proven PT69 single board computer. Powerful performance • reliability • an unbeatable combination! Complete Systems feature:

- Double Sided/Double Density Drives (40 or 80 track) Cabinet, Power Supply
- Optional CRT and Printer
- 1MHz 6809E Processor
- 2 RS232 Serial Ports
- 2 8-Bit Parallel Ports
- 56K RAM: 4K EPROM: 4K I/O

\* COMPLETE SYSTEM with PT69 Board, 2 DS/DD 5 1/4" 40 Track Drives, Cabinet, and Power Supply. \$999.95

\* PT69, Assembled, Tested, with Power Supply and Cabinet. \$399.95

\* PT69, Assembled and Tested. \$299.95

— Kits are available —

— Write for information on CRT and Printer —

### PERIPHERAL TECHNOLOGY

"Supplying Your Computer Needs Since 1978"  
3760 Lower Roswell Road  
Marietta, Georgia 30067

VISA/MASTERCARD/CHECK/COD 404/973-0042

# ElectroScreen™ BT-1

**Single Board Graphics & Text Processor**  
**Now With Complete 2-D Software Package**

David Hanon in 68 Micro Journal —  
"The hardware is very professional,  
... the manual is unusually clear ..."  
(Nov. 1983)

Dale Puckett in InfoWorld  
"... BT-1 is an excellent tool ..."  
(May 23, 1983)

• BT-1 is used by Fermilab, NOAA, GM, NASA, and many other research centers and universities.

### The ElectroScreen™ Intelligent Graphics Board Features:

#### Terminal

- Terminal emulation on power-up
- 84 characters by 48 lines display
- Easy switching between user-defined character sets
- Fast hardware scrolling
- Text qualities such as overstrike, underline, line wrap, reverse video, etc.
- TV-950 graphic character emulation

#### Additional Features

- SS-50 and SS-64 compatible board
- Board communicates with host through parallel latches
- Easy interface to non SS-50 computers via simple parallel port — complete instructions provided
- Composite and TTL level video output
- 8 channel 8 bit A/D converter
- Board occupies only 4 address bytes
- OS-9 level I & II and FLEX drivers available

BT-1 with Version 1.0 software — \$595.

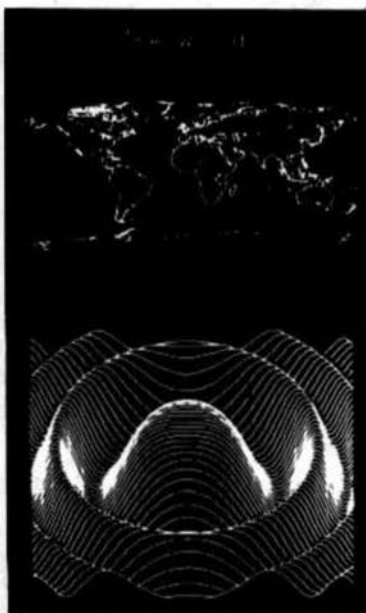
BT-1 with Version 2.0 software (includes complete 2-D package) — \$795.

#### Graphics

- 512 × 480 resolution bit mapped display
- Interleaved memory access-fast, snow-free updates
- Graphic commands sent in simple ASCII characters
- Absolute coordinate or relative cursor moves
- STD syntax high level graphics command set — now includes circles, arbitrarily bounded flood, solid rectangles, variable sized auto-clipping, viewports, etc.

#### Intelligence

- 6809 on-board mpu
- 6K on-board firmware
- Flexible text and graphics integration
- Multiple character sizes
- User programs can be run on-board
- Loads user defined alternate character sets



The Electro Screen manual is available for \$25 credited toward purchase of the board.

The ElectroScreen has a 90 day warranty.



**Privac Inc** (703) 671-3900  
3711 S. George Mason Dr., Falls Church, Va. 22041

**6809 Word Processing System**

# *stylograph*<sup>TM</sup>

**STYLOGRAPH 2.0**  
The "User Friendly" word processing system. Fewer key strokes by the operator make it easier to learn.  
OS9, FLEX \$295 UNIFLEX \$395  
COLOR COMPUTER FLEX \$195

**SPELLING CHECKER**  
Checks all words against an internal user-expandable dictionary of over 42,000 words.  
OS9, FLEX \$145 UNIFLEX \$195

**MAIL MERGE**  
Inserts names and addresses into form letters and mailing lists. Appends files at print out time. Handles files longer than memory.  
OS9, FLEX \$125 UNIFLEX \$175

**INQUIRE ABOUT OUR OTHER SOFTWARE**  
• Business Programs - G/L, A/R, A/P  
• Data Base Management System  
• Assemblers  
Also, Daisy Wheel Printers \$599.

**Great Plains Computer Company Inc.**  
P.O. Box 916  
Idaho Falls, Idaho 83401  
(208) 529-3210

Flex and Uniflex are trademarks of Technical Systems Consultants, Inc.  
OS9 is a trademark of Microware.

**NEW!  
DRIVES**

**\$159.00**

WITH CASE &  
POWER SUPPLY  
\$189.95



TANDON MP3 TEAC

Speed 5mg 10 to 11  
Capacity 250K unformatted  
Tracks 40  
Warranty 6 months

ORDER TOLL FREE!  
1-800-635-0300

**CALL FOR  
SALE  
PRICES**

Even more savings!! ALL DRIVES FULLY TESTED & WARRANTED

Complete Disk Drive with Power Supply & Case .....	\$189.95
Two Drives in Dual Case & Power Supply .....	\$358.00
1/2 hi double sided double density Disk Drives (300000) .....	\$239.95
1/2 hi double sided double density Disk Drive with OS 9/400 .....	\$289.95
Single ps & case .....	\$39.95
Dual ps & case .....	\$69.95
Color Computer Controller Supports double sided double density drives with 1 year warranty and manual .....	\$129.95
Ask about our 3way Color Computer Drive Kits	
Complete with Diskettes .....	starting at \$329.95
Diskettes with 5 1/4 inch format .....	\$19.95
Head Cleaner Kits .....	\$Call
Unadvertised Specials .....	\$Call
Drives cleaned, aligned & tested .....	\$29.95

TECHNICAL STAFF ON DUTY PLEASE CALL FOR ASSISTANCE

**CALL US TODAY! (617) 234-7047** \*DEALER INQUIRIES INVITED



**TRUE DATA PRODUCTS**

100 Linnwood Street, P.O. Box 300  
Lynnwood, Massachusetts 01529  
(617) 234-7047  
HOURS: MON - SAT 9 - 6 (EST)

By MasterCard

\* Visa/Master Charge

\* Check/Money order 2 weeks for clearing

\* C.O.D. Add \$2.00

## QMM2 256K DYNAMIC MEMORY

A 2Mhz dynamic memory board compatible with SWTPC and other systems, Socketed, assembled, tested, gold connectors, 1 yr. warranty. **\$795.00**

## OS-9<sup>TM</sup> SOFTWARE

**UTILITIES-A** for OS-9 level II includes a fast function PRINTERR, disassembler memory and process dumps, file split and others. **\$75.00**

**FILTER KIT #1** Contains 11 utilities used as filters to allow wild card directory lists, copies, deletes, moves, lists, pagination, etc. **\$40.00**

**Terms:** Prepaid, COD, VISA, MASTERCARD.  
Specify 5 1/4" or 8" diskette for software.  
Send SASE for more info.

**D.P. Johnson (503) 244-8152**  
7655 S.W. Cedarcrest St., Portland, OR 97223  
OS-9 is a trademark of Microware and Motorola Inc.

# OS-9<sup>®</sup>

## SOFTWARE TOOLS!

- **ADLIB 50.00**  
Allows shared source code
- **ISAM 350.00**  
Indexed Sequential File Access Method
- **SMATH 95.00**  
String Arithmetic
- **SORTC 150.00**  
Full-record disk sort
- **EXAMOD/CHGREV 50.00**  
Determines contents of a module
- **IN-DATA/REPORT-GEN/MENU 75.00**  
Data entry/Menu & Report generator
- **LOOKUP/SLOOKUP 75.00**  
"Wild Card" directory searcher
- **VID 75.00**  
Display/input data edit package

- **XRF 200.00**  
Low-overhead database emulator
- **TERMINAL 95.00**  
Communicates with other machines
- **LDMAC 75.00**  
Assembly code routines "tool box"

For more information, or to place an order,  
contact:

Dept. 68 8

The JBM Group, Inc.

**the JBM group**

Continental Business Center  
Front & Ford Streets  
Bridgeport, PA 19405  
Tel: 215-337-3138/TWX: 510-660-3999

VISA/MC accepted: PA residents please add 6% sales tax  
U.S. orders please add 5.00 postage/handling

® Registered trademark of Microsoft

### 68 MICRO JOURNAL PROGRAMS on DISK

Disk #1: FILESORT, MINICAT, MINICOPY, MINIFMS,  
\*\*LIFETIME, \*\*POETRY, \*\*FOODLIST, \*\*DIET.  
Disk #2: DISKEDIT w/ inst. & fixes, PRIME, \*PRMOD,  
\*\*SNOOPY, \*\*FOOTBALL, \*\*EXPAWN, \*\*LIFETIME.  
Disk #3: CBUG09, SEC1, SEC2, FIND, TABLE2, INTEXT,  
DISK-EXP, \*DISKSAVE.  
Disk #4: MAILING PROGRAM, \*FINDDAT, \*CHANGE,  
\*TESTDISK.  
Disk #5: \*DISKFIX 1, \*DISKFIX 2, \*\*LETTER,  
\*\*LOVESIGN, \*\*BLACKJAK, \*\*BOWLING.  
Disk #6: \*\*PURCHASE ORDER, INDEX (Disk file Indx).  
Disk #7: Linking Loader & RLOAD, Harkness  
Disk #8: CRTSET, Lanpher (May '82)  
Disk #9: DATECOPY, DISKFIX9 (Aug '82)

NOTE: All are as published or received by 68  
Micro Journal, some have fixes and patches.

This is a reader service only! No Warranty is  
offered or implied, they are as received and are  
for reader convenience ONLY. Also 6800 and 6809  
programs are mixed, as each is fairly simple  
(mostly) to convert to the other.

PRICE: 8" Disk \$19.95 - 5" Disk \$17.95

68 MICRO JOURNAL  
POB 794  
Hixson, TN 37343  
615-642-4600

\* Indicates 6800, \*\* Indicates BASIC SWTPC or  
TSC - 6809 no Indicator.

MASTER CARD - VISA accepted - Foreign add  
sufficient postage surface or air!!

### ANDERSON COMPUTER CONSULTANTS & Associates

Ron Anderson, respected author and columnist  
for 68 MICRO JOURNAL announces the **Anderson  
Computer Consultants & Associates**, a con-  
sulting firm dealing primarily in 68XX(X)  
software design. Our wide experience in  
designing 6809 based control systems for  
machine tools is now available on a  
consultation basis.

Our experience includes programming  
machine control functions, signal analysis,  
multi-axis servo control (CNC) and general  
software design and development. We have  
extensive experience in instrumentation and  
analysis of specialized software. We support  
all popular languages pertaining to the 6809  
and other 68XX(X) processors.

If you are a manufacturer of a control or  
measuring package that you believe could  
benefit from efficient software, write or call  
Ron Anderson. The fact that any calculation  
you can do with pencil and paper, can be done  
much better with a microcomputer. We will be  
happy to review your problem and offer a  
modern, state-of-the-art microcomputer  
solution. We can do the entire job or work  
with your software or hardware engineers.

**Anderson Computer Consultants & Associates**  
3540 Sturbridge Court  
Ann Arbor, MI 48105



# IN CASE OF OS9\* EMERGENCY

Computer &  
Communication  
Specialists

the **JBM**  
group

## OS9 SOLUTION TEAM

(215) 337-3138  
TWX-510-660-3999

Dept. 68 7  
Continental Business Center  
Front & Ford Streets  
Bridgeport, PA 19405

# BREAK GLASS

OS9 is a registered trademark of Microware Corporation.

# FREE CATALOG

**FHI FRANK HOGG LABORATORY**  
THE REGENCY TOWER • SUITE 215 • 770 JAMES ST. • SYRACUSE, NY 13203  
PHONE (315) 474-7856 • TELEX 848740

## ARCADE 50

**POWERFUL COLOR GRAPHICS**  
Uses the new TMS9918A Video Display processor. High resolution 256 x 192 pixel display with 15 colors. 16K Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation. External Video input allows subtitling. NTSC composite video output.  
**SOUND EFFECTS AND MUSIC**  
• Three AY3-8910 Programmable Sound Generators  
• Nine simultaneous voices  
• Three independent noise sources  
• Onboard stereo amplifier drives two 8 ohm speakers

**ADDITIONAL I/O CAPABILITIES**  
• Eight analog inputs with 8 bit resolution  
• Supports four joysticks with pushbutton switches  
• Eight bit parallel I/O port  
• Entire unit map into 256 bytes of memory

## FBASIC

TERMINUS DESIGN INC. in conjunction with Microware Systems Corporation, is proud to announce FBASIC, an enhancement of Microware's 6800/BASIC. Their fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions, Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50.

ARCADE 50 assembled and tested	\$325.00
Video and Audio connector set	15.00
4 Joystick connector set	15.00
2 Radio Shack Joystick	24.00
Gold Molex connectors	12.00
A/BASIC for 6800	110.00
FBASIC for 6809	110.00
FBASIC (with ARCADE 50)	75.00
ARCADE 50 RGB	375.00
LABVIDEO (Motorola EXORbus)	375.00
NEW M 09 6809 Processor Board	225.00
256K Dynamic Memory Board	795.00
256K Dynamic Memory Board (w/64K)	395.00
64K Dynamic Memory Board	295.00

TERMINUS DESIGN INC  
16 SCARBROUGH ROAD  
ELLENWOOD, GA 30049  
(404) 474-4866

TERMINUS DESIGN, INC. (404)

## Computer Servo Controlled Robot Arm



Call or Write for Free Catalog

**Analog Micro Systems**

5660 Volmont Road • Boulder, Colorado 80301 • Tel: (303) 444-6809

## Robot-1

Keyboard or Joystick Control

**Remembers Everything It Did  
& does it again**

**Typical System Includes:**

- Robot-1 & Cables
- 6 Channel Servo Controller
- Power Supply
- All Software with source code

**Modular Robotic Accessories:**

- Mobile Cart for Travelling Robot
- Radio Links between all Functions
- Robot-mounted MicroEye
- Ultrasonic Range Finder

Robot-1C for Color Computers - \$395.00

Robot-1S for S850 Systems - \$395.00

Robot MicroEye - \$295.00

**Additional Systems Available**

Robot-10 for General Purpose Computers

Robot-1R for Radio Control Systems



#### MODULES - BARE CARDS - KITS - ASSEMBLED & TESTED

Stackable Modules	KIT	A&T
20 amp POWER SUPPLY w/fan w/Disk protect relay	350.00	400.00
DISK CABINET w/reg. & cables less DRIVES	200.00	250.00
MOTHER BOARD, 8 SS-50c, 8 SS-30c MMI button	225.00	325.00

Item	Bare	KIT	A&T
ITS - INTERRUPT TIMER 1, 10, 100 per sec. 19.95	29.95	39.95	
PB4 - INTELLIGENT PORT BUFFER Single board comput. 39.95	114.95	139.95	
DP1A - Dual PIA parallel port, 4 buffered I/Os 24.95	69.95	89.95	
XADR - Extended Addressing BAUD gen. PIA port 29.95	69.95	89.95	
MB8 - MOTHER BOARD SS-3 c w/BAUD gen. 64.95	149.95	199.95	
P168 - 168K PROM DISK 21, 2764 EPROMs 39.95	79.95	109.95	
FD88 - Firmware development 2, 8K blocks 39.95	84.95	114.95	
XMPR - 2764 PROM burner adapt. for 2716 BURNER 19.95	19.95		
CHERRY Keyboard w/Cabinet 96 key capacitive 249.95			
TAKAN 12", 18 Mhz MONITOR GREEN AMBER 149.95			159.95
4 MODULE CABINET - unfinished 150.00			
POWER SUPPLY w/disk protect 250.00			

## Color Computer

WOPOLINE - 20 Mhz Monochrome video driver 15.00	20.00
CC30 PORT 808 w/power supply 5 SS-30, 2 Cart 169.95	199.95
POWER BOX 8 switched outlets transient suppression 29.95	39.95
RS-232 3-switched ports for above ADD +20.00	+25.00

Write for FREE Catalog

ADD \$3.00 SEM PER ORDER  
WIS. ADD 5% SALES TAX



11931 W. Bluemound Road  
MILWAUKEE, WIS. 53226  
(414) 257-0300

## '68' MICRO JOURNAL ADVERTISERS INDEX

'68' MICRO JOURNAL	50,67
AAA CHICAGO COMPUTER CENTER	36,37
ACORN COMPUTER SYSTEMS	70
ANALOG MICRO SYSTEMS	69
ANDERSON COMPUTER CONSULTANTS	67
COLOR MICRO JOURNAL	6
COMPILER EVALUATION SERVICES	49
COMPUTER PUBLISHING INC.	5
COMPUTER SYSTEMS CENTER	59,60
COMPUTER SYSTEMS CONSULTANTS, INC.	51
D.P. JOHNSON	66
DATA-COMP	48,1BC
DIGITAL RESEARCH COMPUTERS	62,63
FRANK HOGG LABORATORY, INC.	69
GIMIX, INC.	3,72
GREAT PLAINS COMPUTER CO.	66
HAZELWOOD COMPUTER SYSTEMS	60C
INTROL CORP.	50
JBM	67,68
JOTO ASSOCIATES	71
LSI ENTERPRISES LTD.	58
MACROPLEX SOFTWARE	64
MICROWARE SYSTEMS CORP.	1,4,12
PERIPHERAL TECHNOLOGY	64,65
PRIVAC INC	65
ROBERTSON ELECTRONICS	58
SNOKE SIGNAL BROADCASTING	7
SOUTH EAST MEDIA	52,53,54,55,56,57
SOUTHWEST TECHNICAL PRODUCTS INC.	1FC
STAR-KITS	49
TALBOT MICROSYSTEMS	51
TERMINUS DESIGN, INC.	69
THE VIRGINIA COMPANY	71
THOMAS INSTRUMENTATION	71
TRUE DATA PRODUCTS	56
UNITIK	58
WESTCHESTER APPLIED BUSINESS SYSTEMS	64
WINDRUSH MICRO SYSTEMS LIMITED	61

This index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

## THOMAS INSTRUMENTATION

### HARDWARE

CPU-5  
S-R/R w/o memory chips  
S-R/R with 48K  
SP-1  
6802 SUPER CPU  
VIDEO RAM  
PARALLEL I/O  
SS-50/50C EXTENDER  
SS-30 EXTENDER  
SS-50 WIRE WRAP  
SS-30 WIRE WRAP  
SS-30 BACKPLANE 8POS.  
SS-50 BACKPLANE 4,8,12,  
& 16 @ \$5.00/SLOT

### ASSEM & TEST

\$299.00  
\$120.00  
\$399.00  
\$195.00  
\$235.00  
\$195.00  
\$139.00  
\$ 35.00  
\$ 25.00  
NA  
NA  
NA  
NA

### BARE CARD

\$49.00  
\$49.00  
\$49.00  
\$49.00  
\$59.00  
\$49.00  
\$49.00  
NA  
NA  
\$39.00  
\$20.00  
\$39.00  
NA

### SOFTWARE

VDISK  
Use extended memory  
as a fast disk drive  
6809 source & obj \$149.00  
6809 object \$ 99.00  
  
OUTSIDE MODEM PROGRAM  
includes source  
UniFLEX™ \$100.00  
FLEX 6800 or 6809 \$ 50.00  
  
CROSS ASSEMBLER  
For 6800, 6801, 6805  
Runs on 6809 FLEX™  
\$150.00

### COMPONENTS

GOLD MALE MOLEX \$1.60 GOLD FEMALE MOLEX \$1.60  
TIN MALE MOLEX \$ .40 TIN FEMALE MOLEX \$ .50  
146805E2P \$20.00

THOMAS INSTRUMENTATION

168 EIGHTH STREET

AVALON, N.J. 08202

(609) 967-4280

MASTERCARD, VISA, AND C.O.D. ACCEPTED. PLEASE ALLOW 3 WEEKS  
FOR CHECKS TO CLEAR. CONT. USA ADD \$1.00 SHIPPING. CANADA  
\$6.00. FOREIGN \$12.00. NJ RESIDENTS ADD 6% SALES TAX  
FLEX and UniFLEX ARE TRADEMARKS OF TECHNICAL SYSTEMS CONSULTANTS

## *Bizpack*

welcomes you to

Business planning, budgeting, and forecasting. One of the most useful programs you can find on any computer, any size, any price. More than just a spread sheet. Flex and TSC Xbasic required. With source, License \$250 (subject to non-disclosure agreement). Standard *Bizpack* License \$135.

**The Virginia Company**

P.O. Box 2167 Christiansburg, Virginia 24073 • (703) 382-4135

## FIRST CLASS

### INTELLICOM

INTELLigent COMmunications Program

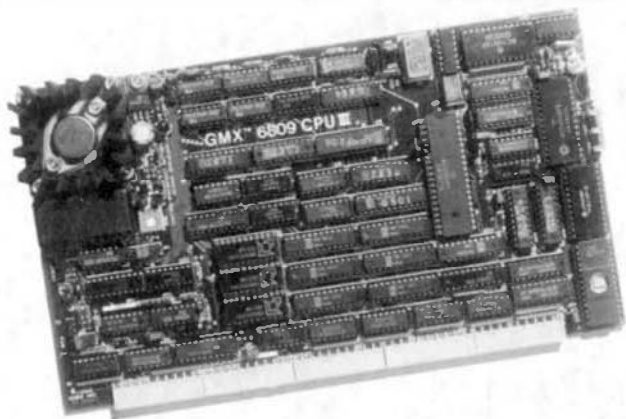
INTELLICOM provides you with the capability of intelligent computer to computer communications from both a terminal emulation and file transfer standpoint. INTELLICOM supports several file transfer protocols that facilitate the transfer of both binary and ASCII data. Since INTELLICOM is menu drive, it is a breeze to use and understand. With INTELLICOM you will be able to communicate with the various data and timesharing services such as The Source and CompuServe. Additionally, since INTELLICOM supports the protocol used by virtually all remote CP/M systems around the world, all users can immediately begin to take advantage of the wealth of public domain software available on these systems. Current, or potential, users of CompuServe can transmit and receive both binary and ASCII data with full error detection and recovery. The checksum protocols allow for the verifications of data blocks transferred (assuming appropriate support on the host end). This feature will be of great value in those applications where data integrity is paramount. INTELLICOM's documentation includes a detailed description of all protocols used along with machine readable examples of host pseudo code. A telephone directory and dialing facility compatible with Hayes Smartmodems is supported.

\*FLEX is a trademark of Technical Systems Consultants  
\*OS9 is a trademark of Microsoft

Price: OS9 or FLEX Version \$99.95, add \$5.00 COD  
JOTO ASSOCIATES, 104 Lehigh Dr., Southington, CT 06489

203-621-8070

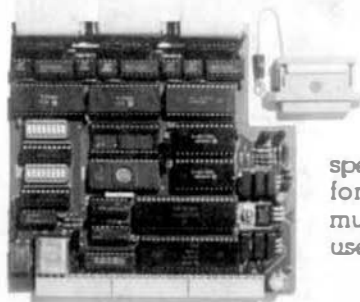
# GIMIX STATE OF THE ART 6809 SYSTEMS FOR THE SERIOUS USER.



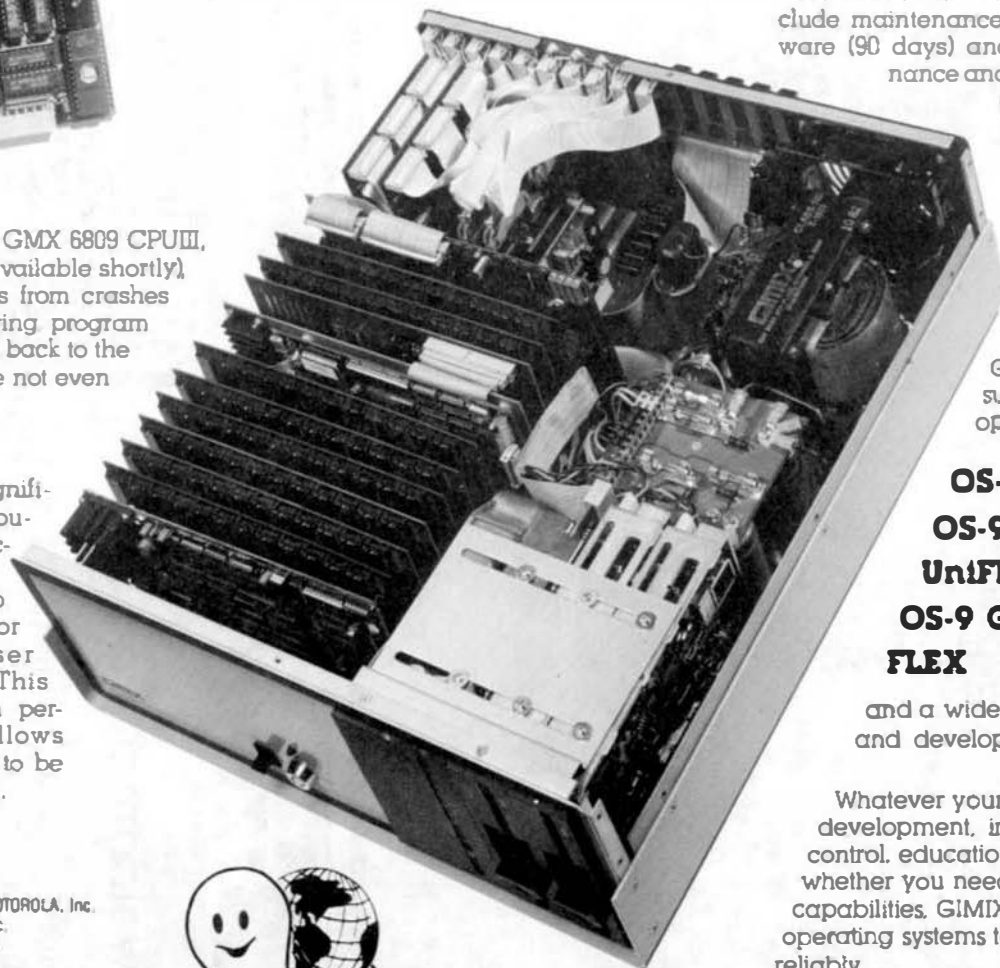
**GIMIX has 19MB or high performance 47MB Winchester Drive Systems and/or Floppy Disk Drive Systems.**

For the ultimate in performance, the Unique GMX 6809 CPU III, using either OS-9-GMX III or UniFLEX GMX III (available shortly) gives protection to the system and other users from crashes caused by defective user programs. e.g. During program development, a programmer who crashes goes back to the shell or the debugger, while the other users are not even aware anything occurred.

The intelligent serial I/O processor boards significantly reduce system overhead by handling routine I/O functions, thereby freeing up the host CPU for running user programs. This speeds up system performance and allows multiple terminals to be used at 19.2K baud.



BASIC-09 and OS-9 are trademarks of Microware Systems Corp. and MOTOROLA, Inc. FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc. GIMIX, GHOST, GMX, CLASSY CHASSIS, are trademarks of GIMIX, Inc.



GIMIX 6809 systems support five predominant operating systems:

**OS-9 GMX III,  
OS-9 GMX II,  
UniFLEX,  
OS-9 GMX I,  
FLEX**

and a wide variety of languages and development software.

Whatever your application: software development, instrumentation, process control, educational, scientific or business, whether you need single or multi-user capabilities, GIMIX has hardware and the operating systems to get the job done reliably.

Please phone or write if you need further information.



## GIMIX inc.

1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609 • (312) 927-5510 • TWX 910-221-4055

©1983 GIMIX Inc.



# The Original FLEX™ for Color Computers

- Upgrade to 64K
- RS to FLEX, FLEX to RS file transfer ability
- Create your own character set
- Automatic recognition of single or double density and single or doubled sided
- All features available for either single or multiple drive systems
- Settable Disk Drive Seek Rates
- Faster High Resolution Video Display with 5 different formats
- Save RS Basic from RAM to Disk
- Move RS Basic to RAM
- Load and save function on FLEX disk
- 24 Support Commands 12 with Source Text
- External Terminal Program

DATA-COMP has everything you need to make your TRS-80C Color Computer WORK for YOU; from Parts and Pieces to Full Ready To Use Systems. DATA-COMP designs, sells, services, and SUPPORTS Computer SYSTEMS, not just software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

## System Requirements

**FLEX** Special General Version + Editor & Assembler (which normally sell for \$50.00 ea.) \$250.00  
**F-4025(RS)** FLEX Conversion Rout. for the RS Disk Controller when purchased with Special General FLEX Sys. \$ 49.95  
 when purchased without the General FLEX Sys. \$ 59.95

Color Computer with 64K RAM and EXT.BASIC \$399.95

## SPECIAL SYSTEM PACKAGES

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General Version of FLEX, F-4025(RS), and a Box of 10 Double Density Diskettes, a COMPLETE ready to run SYSTEM on your Color TV Set.

\$1899.95

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, and a Box of 10 Double Density Diskettes, a COMPLETE ready to run SYSTEM on your Color TV Set.

\$799.95

## PARTS AND PIECES

Radio Shack Disk Controller \$169.95  
 1 Single Sided, Double Density Disk Drive Tandom \$229.95  
 1 Double Sided, Double Density Disk Drive Qume \$319.95  
 1 Qume Thinline Double Sided, Double Density \$249.95  
 1 Tandom 40 Track SSD0 with Cabinet and Power Supply \$259.95  
 Single Drive Cabinet with Power Supply \$ 79.95  
 Double Drive Cabinet with Power Supply \$109.95  
 J & M Disk Controller \$144.95

Single Drive Disk Cable for RS Controller \$ 19.95  
 Double Drive Disk Cable for RS Controller \$ 24.95  
 Micro Tech. Prods. Inc. LOWER CASE ROM Adapter \$ 74.95  
 Radio Shack BASIC Version 1.1 ROM or 1.2 ROM \$ 34.95  
 Radio Shack Extended Basic ROM \$ 39.95  
 Radio Shack Disk Basic ROM 1.1 \$ 29.95  
 Screen Clean-Clears Up Video Distortion On Your Color Computer \$ 39.95  
 Set of Eight 64K RAM Chips w Mod Instructions \$ 49.95

## DISK DRIVE PACKAGES, etc.

These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable:

PAK #1 - 1 Single Sided, Double Density Sys. \$389.95  
 PAK #2 - 2 Single Sided, Double Density Sys. \$675.95  
 PAK #3 - 1 Double Sided, Double Density Sys. \$569.95  
 PAK #4 - 2 Double Sided, Double Density Sys. \$889.95  
 PAK #5 - 2 Qume Thinline Double Sided, Double Density Sys \$699.95  
 PAK #6 - 2 Tandom Thinline Single Sided, Double Density Sys \$599.95

## PRINTERS

EPSON RX-80 \$325.00  
 EPSON RX-80FT \$375.00  
 EPSON MX-100 \$675.00  
 EPSON FX-100 \$799.00  
 EPSON FX-80 \$549.00  
 EPSON MX-70 \$230.00

## SERIAL BOARDS

MC-Series \$119.95  
 RX-FX-Series \$ 99.95  
 Spectrum MC-Series \$ 54.95

H J L Keyboards \$ 74.95

## FLOPPY DISKETTES

### MEMOREX VERBATIM

5" Soft Sector Disks  
 Single Sided Single Density \$2.40ea. \$2.75ea  
 Single Sided Double Density \$2.40ea \$2.75ea  
 Double Sided Double Density \$4.92ea  
 Plastic Storage Box \$2.00ea  
 8" Soft Sector Disks  
 Single Sided Single Density \$3.75ea  
 Single Sided Double Density \$4.10ea  
 Double Sided Double Density \$4.75ea  
 Plastic Library Box \$5.00ea

For Ordering Call TOLL FREE 1-800-338-6800

**LAST OF PRODUCTION!!!**

**!! LIMITED QUANTITY !!**

**SWTPC 8212 & 8212(W)**  
**Intelligent Terminals**

**New & Demo Models**

**At Discount Prices**

**Remaining Supply of**  
**SWTPC 8212 CRT Terminals**

**EM-82 Video**  
**Terminal**

Emulates the 8200 Series from SWTPC



\*FLEX is a trademark of Technical Systems Consultants  
 \*OSB is a trademark of Microware

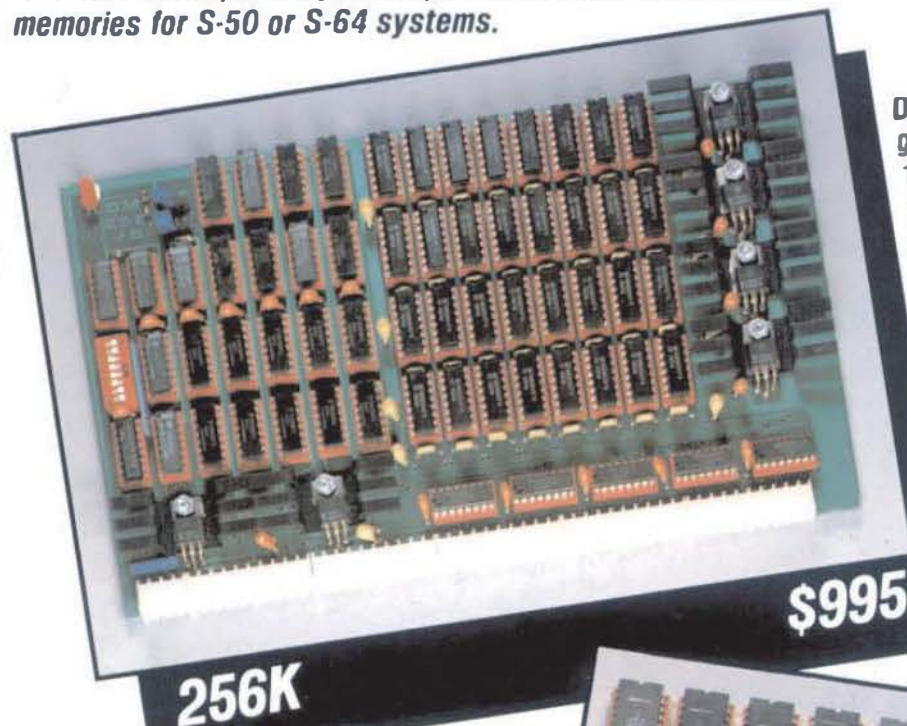
Data-Comp—South East Media & 68 Micro Journal Are Divisions of CPI



000422 A/E  
MR. MICKEY FERGUSON  
P.O. BOX 87  
KINGSTON SPRINGS TN 37082

# MEMORY • MEMORY • MEMORY • MEM

Hazelwood Computer Systems presents a pair of advanced memories for S-50 or S-64 systems.



**\$995**

**256K**

## Dynamic memory with guaranteed 2.5 MHz operation

The DM-256 utilizes the same proven refresh method as our original DM-64. This technique allows fully transparent refresh while maintaining full bus speed and eliminating difficult timing problems.

In addition, individual power regulators for each row of memory ICs enhance reliability by providing the ultimate in noise decoupling. Flexible addressing mode controls make the entire 256K available to OS9 Level 2 or UniFLEX. For FLEX, the furnished utility program HYPERDSK™ formats memory not used by FLEX as a virtual, superfast disk, thus tremendously improving total system performance.

With several hundred already in use, the DM-256 is the best choice for economical, reliable memory.

Assembled, tested, and burned in  
Order: DM-256



**64K**

**\$395**

## Static memory with on-board battery backup

Utilizing the latest in low power CMOS static RAMs, the CM-64 has an on-board NiCad battery to maintain memory contents in the absence of bus power. Generous timing margins allow operation at bus speeds in excess of 2 MHz. Zener diode power loss sensing inhibits write operations to prevent data loss during power transitions. Write access can be explicitly controlled by means of an on-board jumper, a remote mounted switch, or an external logic signal.

This allows the memory to appear as ROM to the computer after the contents have been "programmed" using normal computer operations. The economical price makes the advantages of non-volatile memory available to any system.

Assembled, tested, and burned in  
Order: CM-64

## HAZELWOOD COMPUTER SYSTEMS

907 East Terra, O'Fallon, MO 63366,

314-281-1055

**HELIX**